

Automatic Image Captioning with Style

Alexander Mathews

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

November 2018

Except where otherwise indicated, this thesis is my own original work.

A handwritten signature in black ink, reading "Alex Mathews". The signature is written in a cursive style with a large, stylized 'A' and 'M'.

Alexander Mathews

1 November 2018

to my parents Ann and Richard, and my brothers Jamie and Max

Acknowledgments

First, I would like to express my thanks to Dr. Lexing Xie, and Dr. Xuming He for being such excellent supervisors. I could not have completed this thesis without your constant guidance and input. Your enthusiasm is contagious. Thank you.

To the members of the ANU Computational Media group, thank you for the insightful discussions and companionship over the past few years.

I would also like to thank my parents, Ann Beveridge and Richard Mathews, for their support throughout my PhD. For proofreading this thesis I have Ann Beveridge to thank; the sheer number of commas added was a sight to behold. A special thanks to my partner Joanna Howes for her constant encouragement, support, and positive outlook.

This research is supported by an Australian Government Research Training Program (RTP) Scholarship, Data2Decision CRC, and Nvidia who provided a GPU through the hardware grant program. I thank these organisations for their support.

Abstract

This thesis connects two core topics in machine learning, vision and language. The problem of choice is image caption generation: automatically constructing natural language descriptions of image content. Previous research into image caption generation has focused on generating purely descriptive captions; I focus on generating visually relevant captions with a distinct linguistic style. Captions with style have the potential to ease communication and add a new layer of personalisation.

First, I consider naming variations in image captions, and propose a method for predicting context-dependent names that takes into account visual and linguistic information. This method makes use of a large-scale image caption dataset, which I also use to explore naming conventions and report naming conventions for hundreds of animal classes. Next I propose the SentiCap model, which relies on recent advances in artificial neural networks to generate visually relevant image captions with positive or negative sentiment. To balance descriptiveness and sentiment, the SentiCap model dynamically switches between two recurrent neural networks, one tuned for descriptive words and one for sentiment words. As the first published model for generating captions with sentiment, SentiCap has influenced a number of subsequent works. I then investigate the sub-task of modelling styled sentences without images. The specific task chosen is sentence simplification: rewriting news article sentences to make them easier to understand. For this task I design a neural sequence-to-sequence model that can work with limited training data, using novel adaptations for word copying and sharing word embeddings. Finally, I present SemStyle, a system for generating visually relevant image captions in the style of an arbitrary text corpus. A shared term space allows a neural network for vision and content planning to communicate with a network for styled language generation. SemStyle achieves competitive results in human and automatic evaluations of descriptiveness and style.

As a whole, this thesis presents two complete systems for styled caption generation that are first of their kind and demonstrate, for the first time, that automatic style transfer for image captions is achievable. Contributions also include novel ideas for object naming and sentence simplification. This thesis opens up inquiries into highly personalised image captions; large scale visually grounded concept naming; and more generally, styled text generation with content control.

Publications, Software & Data

The following original publications were made during the development of this thesis as part of the Doctor of Philosophy programme. Software and data produced for these publications is provided so that it may form a basis for future work.

Publications

Mathews, A.; Xie, L. & He, X., SemStyle: Learning to Generate Stylised Image Captions using Unaligned Text, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018

<https://arxiv.org/abs/1805.07030>

Mathews, A.; Xie, L. & He, X., Simplifying Sentences with Sequence to Sequence Models, *arXiv preprint*, 2018

<https://arxiv.org/abs/1805.05557>

Mathews, A.; Xie, L. & He, X., SentiCap: Generating Image Descriptions with Sentiments, *Thirtieth AAAI Conference on Artificial Intelligence*, 2016

<https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12501>

Mathews, A., Captioning Images Using Different Styles, *MM'15 ACM Multimedia Conference: Doctoral Symposium*, 2015

<https://dl.acm.org/citation.cfm?id=2807998>

Mathews, A.; Xie, L. & He, X., Studying Object Naming with Online Photos and Caption, *MM'15 ACM Multimedia Conference: MMCommons Workshop*, 2015

<https://dl.acm.org/citation.cfm?id=2814817>

Mathews, A.; Xie, L. & He, X., Choosing Basic-Level Concept Names Using Visual and Language Context, *Winter Conference on the Applications of Computer Vision (WACV)*, 2015

<https://ieeexplore.ieee.org/abstract/document/7045939/>

Software & Data

SemStyle: source code, dataset, models, and evaluation results.

<https://github.com/computationalmedia/semstyle>

SentiCap: source code, dataset, and models.

<http://cm.cecs.anu.edu.au/post/senticap/>

Choosing Names with Context: comparison data, name catalogue, and results.

<https://github.com/computationalmedia/naming-with-visual-context>

Contents

Acknowledgments	vii
Abstract	ix
Publications, Software & Data	xi
1 Introduction	1
1.1 Main Research Challenges	3
1.1.1 Style and Content Representation	3
1.1.2 Generative Models for Styled Captions	4
1.1.3 Data Scarcity	5
1.2 Thesis Outline	6
1.3 Main Contributions	8
2 Related Work	11
2.1 Object Recognition	12
2.1.1 Neural Networks for Classification	14
2.1.2 Convolutional Neural Networks	16
2.2 Language Generation and Recurrent Neural Networks	20
2.2.1 Recurrent Neural Networks	22
2.2.2 Sequence-to-Sequence Models	25
2.2.2.1 Attention	26
2.3 Image-Caption Generation	27
2.3.1 Caption Retrieval	28
2.3.2 Object Detectors and Text Generators	30
2.3.3 End-to-End Neural Network Models	32
2.3.4 Evaluating Generated Captions	34
2.4 Defining Style	37
2.4.1 Sentiment	38
2.4.2 Style Usage and Effect	40
2.5 Separating Style from Content	41
2.5.1 Topic Models	43
2.5.2 Authorship attribution	44

2.5.2.1	Features	45
2.6	Styled Text Generation	46
2.7	Style Transfer for Text	48
2.7.1	Adversarial Approaches	48
2.7.2	Approaches Requiring Sentence Alignment	49
2.7.3	Other Approaches	50
2.8	Caption Generation with Style	51
2.8.1	Captions with Sentiment	52
2.8.2	Multilingual Captioning	54
2.9	Summary	54
3	Object Naming for Image Captions	57
3.1	Introduction	57
3.2	Background	58
3.3	Object Naming in Context: A Pilot Study	61
3.3.1	Dataset and Pre-processing	62
3.3.2	Model	62
3.3.3	Learning Set-up	63
3.3.4	Results	64
3.4	Large Scale Object Naming with Visual Context	65
3.4.1	Method	66
3.4.1.1	Detecting Visual Concepts	66
3.4.1.2	Defining a Concept's Name Vocabulary	68
3.4.1.3	Naming Visual Concepts	69
3.4.1.4	Ranking Names and Concepts	70
3.4.2	Experimental Setup	72
3.4.2.1	Training and Testing Datasets	72
3.4.2.2	Model Learning	73
3.4.2.3	Evaluation Metrics and Baselines	73
3.4.3	Results	74
3.4.3.1	Name from Visual Context with a Known Concept	74
3.4.3.2	Image to Names	77
3.5	Large Scale Naming Patterns in a Taxonomy	77
3.5.1	Datasets and Pre-processing	78
3.5.2	Model	79
3.5.3	Results	79
3.5.3.1	Large Scale Naming Patterns	80
3.5.3.2	Human Evaluation	81

3.5.3.3	Concept Detector Visualisations	82
3.6	Summary	83
4	Generating Image Captions with Strong Sentiment	95
4.1	Introduction	95
4.1.1	Overview of the SentiCap System	96
4.2	Related Work	96
4.2.1	Transfer Learning	97
4.2.2	Sentiment	98
4.3	Generating Image Captions with Sentiment	100
4.3.1	Model Overview	100
4.3.1.1	Objective Functions	101
4.3.2	Model Component Details	103
4.4	Constructing a Dataset of Captions with Sentiment	106
4.4.1	Adjective Noun Pair Vocabulary Construction	107
4.4.2	Collecting Image Captions with Sentiment	107
4.4.3	Dataset Validation	110
4.5	Experiments	111
4.5.1	Implementation Details	111
4.5.2	Dataset Setup	112
4.5.3	Baselines	112
4.5.4	Evaluation Metrics	113
4.5.5	Results	114
4.6	Summary	119
5	Simplifying Sentences	123
5.1	Introduction	123
5.2	Related Work	124
5.2.1	Sentence Simplification as Machine Translation	125
5.2.2	Related Problems	127
5.2.3	Datasets	130
5.3	Model	131
5.3.1	Sequence to Sequence with Attention	131
5.3.2	Mixing Pre-trained and Trainable Word Embeddings	132
5.3.3	Attentive Word-Copy Feeding	133
5.3.4	Loss Function for Word-Copying	134
5.4	Evaluation Settings	134
5.4.1	Newsela Dataset	134

5.4.1.1	Aligning Sentences	135
5.4.1.2	Manual Word Level Alignment	137
5.4.2	Moses Baseline	137
5.4.3	Evaluation metrics	137
5.5	Results	138
5.5.1	Sequence to Sequence Performance	138
5.5.2	Ablation Study	139
5.5.3	Simplification Examples	140
5.5.4	Attention Alignment Performance	140
5.5.5	Word Replacement Performance	141
5.6	Summary	142
6	Captioning Images with Style Transfer from Unaligned Text Corpora	145
6.1	Introduction	145
6.2	Model	146
6.2.1	Semantic Term Representation	147
6.2.2	Importance Ordering for Parts-of-Speech	151
6.2.3	Generating semantic sequences from images	153
6.2.4	Generating styled descriptions	154
6.3	Learning with Unpaired Styled Texts	155
6.3.1	Training the term generator	155
6.3.2	Training the language generator	156
6.4	Evaluation Setting	156
6.4.1	Datasets	157
6.4.2	Baselines	158
6.4.3	Model Variants	160
6.4.4	Evaluation Metrics	160
6.4.4.1	Automatic Metrics	161
6.5	Results	162
6.5.1	Evaluating Relevance	166
6.5.2	Evaluating Style	166
6.5.3	Human Evaluations	168
6.5.4	Evaluating Modelling Choices	169
6.5.5	Example Captions	170
6.5.6	Coverage of Semantic Terms	170
6.5.7	Diversity	172
6.5.8	Exploring the Generated Style	172
6.6	Summary	174

7	Conclusion	177
7.1	Summary	177
7.2	Future Work	179
7.3	Final Remarks	181

List of Figures

2.1	A simple Recurrent Neural Network. The recurrent layer R is duplicated, to match the length of the input $\{x_1, x_2 \dots x_N\}$ and output $\{y_1, y_2 \dots y_N\}$ sequences.	22
2.2	The StyleNet model [Gan et al., 2017a] for generating styled captions. .	51
2.3	The neural-storyteller model [Kiros, 2015], for generating short styled stories about images. The mean shift block subtracts off the mean skip-thought vector for captions and adds on the mean skip-thought vector for the target style.	52
3.1	Different names can be given to the same concept given a different viewpoint.	59
3.2	We build a classifier that predicts a name for a concept of interest. Ground-truth co-occurring objects are the features, while the caption defines the ground-truth name.	61
3.3	Improvement in name prediction accuracy when context is given compared to no context. All 80 MSCOCO concepts are shown ordered by improvement. See Fig 3.3 for raw accuracy scores.	64
3.4	Name prediction accuracy when context is given (top) compared to no context (bottom). All 80 MSCOCO concepts are shown ordered by delta accuracy improvement from including context. See Fig 3.3 for delta improvements.	65
3.5	Method overview for context-dependent name prediction. See Section 3.4.1 for details.	67
3.6	Per-synset accuracy improvement of <i>BasicName-Visual</i> over the <i>Frequency+described</i> baseline, ordered by accuracy delta. Compare to accuracy in Figure 3.7 (same x-axis order). See Section 3.4.3.1 for discussions. 75	
3.7	Per-synset accuracy for the <i>Frequency+described</i> baseline (top) and <i>BasicName-Visual</i> (bottom), ordered by accuracy delta. Compare to delta accuracy in Figure 3.7 (same x-axis order). See Section 3.4.3.1 for discussions. . .	75

3.8	Examples of context-dependent naming. For each synset we display crowd-sourced one-name-per-synset [Ordonez et al., 2013], n-gram based most frequent name [Ordonez et al., 2013], context-dependent names from <i>BasicName-Visual</i> , and four image examples for each name. For synsets without previous naming results see Figure 3.9.	85
3.9	Examples of context-dependent naming. For each synset we display context-dependent names from <i>BasicName-Visual</i> and four image examples for each name. Unlike Figure 3.8, the synsets in this figure had no previous naming results available [Ordonez et al., 2013].	86
3.10	Precision-recall curves for our method and the four baselines on SBU-1KA and SBU-1KB. Error bars show one standard deviation.	87
3.11	Precision-recall curves on SBU-148K. Error bars show one standard deviation.	87
3.12	Example images from the SBU-1KA and SBU-1KB datasets with Amazon Mechanical Turk labels. We show the top names, predicted by our method, <i>BasicName-Visual</i> , and three baselines. Words printed in green match the hand labelled ground-truth. Our method performs well on the first four images but fails on the last two.	88
3.13	The specificity of names describing mammals. The first row is the most general taxon, while the last is the most specific. Each column is a different animal corresponding to a visual classifier. Darker colours indicate larger counts, with columns normalised. Columns with less than 20 detections were filtered out.	89
3.14	The specificity of names describing birds. Darker colours indicate larger counts, with columns normalised.	90
3.15	The specificity of names describing reptiles. Darker colours indicate larger counts, with columns normalised.	91
3.16	Visualisation of <i>Ursus maritimus</i> (polar bear) images using t-SNE with CNN features. Image border colours represent ground-truth names extracted from captions.	92
3.17	Visualisation of <i>Cygnus atratus</i> (black swan) images using t-SNE with CNN features. Image border colours represent ground-truth names extracted from captions.	93
4.1	An overview of the SentiCap model. The factual language model is on top, the sentiment model is on the bottom. The switch component joins the two models, while the CNN is shared.	101

4.2	Illustration of the switching RNN model for captions with sentiment. LSTM cells are described in Eq 4.9. γ_t^0 and γ_t^1 are probabilities of sentiment switch defined in Eq (4.11) and act as gating functions for the two streams via the element-wise multiply blocks.	105
4.3	One example image with both positive and negative captions written by AMT workers.	108
4.4	Mechanical Turk interfaces and instructions for <i>Collecting</i> sentences with a positive (top) and negative (bottom) sentiment.	109
4.5	Mechanical Turk interface and instructions for validating the dataset. .	110
4.6	Summary of quality validation for sentiment captions. The rows are MSCOCO [Chen et al., 2015], and captions with Positive and Negative sentiments, respectively. <i>Descriptiveness</i> \pm <i>standard deviation</i> is rated as 1–4 and averaged across different AMT workers, higher is better. The <i>Correct sentiment</i> column records the number of captions receiving 3, 2, 1, 0 votes for having a sentiment that matches the image, from three different AMT workers.	111
4.7	AMT interface and instructions for <i>comparative rating</i> of generated sentiment sentences	114
4.8	Example results from sentiment caption generation. Columns a+b: positive captions; columns c+d: negative captions. Background colour indicates the probability of the switching variable $\gamma_t^1 = p(s_t \cdot)$: dark if $\gamma_t^1 \geq 0.75$; medium if $\gamma_t^1 \geq 0.5$; light if $\gamma_t^1 \geq 0.25$. Examples in rows 1 and 2 are successful. Examples in rows 3 and 4 have various semantics or sentiment errors, at times with amusing effects. See Section 4.5 for discussions.	120
5.1	The encoder-decoder with attention for sentence simplification.	133
5.2	The validation performance of <i>S4+gv+bce</i> with different numbers of trainable and pre-trained embeddings. Higher BLEU-4 scores indicate greater simplification precision.	140
6.1	<i>SemStyle</i> distills an image into a set of semantic terms, which are then used to form captions of different styles.	145
6.2	An overview of the <i>SemStyle</i> model. The <i>term generator</i> network (in green) is shown in the lower left. The <i>language generator</i> network is in the upper right (in blue)	147
6.3	An overview of the <i>JointEmbedding</i> model. The two embedding components image embedder (in yellow) and <i>sentence embedder</i> (in red) are shown on the left while the <i>sentence generator</i> (in grey) is on the right. .	160

6.4	A screen-shot of the instructions provided to workers when evaluating the relevance of a caption to an image.	163
6.5	A screen-shot of a single question put to workers during the relevance evaluation task.	163
6.6	A screen-shot of the instructions provided to workers when evaluating the conformance of a caption to the desired style.	164
6.7	A screen-shot of a single question asked of workers in the style evaluation task.	164
6.8	Human evaluations for SemStyle and selected baselines, with error bars showing 0.95 confidence intervals over 10 random splits. (a) descriptiveness measured on a four point scale, reported as percentage of generated captions at each level. (b) style conformity as a percentage of captions: unrelated to the image content, a basic description of the image, or part of a story relating to the image.	165
6.9	Example results, including styled (Story) output from <i>SemStyle</i> and descriptive (Desc) output from <i>SemStyle-coco</i> . Four success cases are on the left (a,b,c,d), and two failures on the right (e,f).	167

List of Tables

3.1	The precision and recall at 5, evaluated on the SBU-1KA and the SBU-1KB datasets. This shows <i>BasicName-Visual+Lang</i> outperforming the Ordonez et al. [2013] approach, <i>Ngram-biased+SVM</i> , in terms of both precision and recall.	76
3.2	Common names selected by AMT workers for each animal. Names are in order from most frequent to least (left to right). The table shows names that occur in at least 10% of cases with a matching name, and with the total count greater than 20. As a result of this filtering some cells are empty.	81
4.1	The six adjectives in the SentiBank and SentiCap vocabularies that apply to the most nouns in the vocabulary. <i>Num. Nouns</i> is the number of nouns to which each adjective applies.	108
4.2	Summary of automatic evaluations for captions with sentiment. Columns: <i>SEN%</i> is the percentage of output sentences with at least one ANP; <i>B-1</i> . . . <i>CIDER_r</i> are automatic metrics as described in Section 4.5; where <i>B-N</i> corresponds to the BLEU- <i>N</i> metric measuring the co-occurrences of <i>n</i> -grams.	115
4.3	Summary of crowd-sourced evaluations for captions with sentiment. Columns: <i>SENTI</i> is the fraction of images for which at least two AMT workers agree that it is the more positive/negative sentence; <i>DESC</i> contains the mean and std of the 4-point descriptiveness score: larger is better. <i>DESCCMP</i> is the percentage of times the method was judged more descriptive, or equally descriptive, as the CNN+RNN baseline.	116
4.4	ANPs for positive sentences generated by <i>SentiCap</i> . Nouns are ordered from most common to least, with only the ten most common shown. Paired adjectives are ordered most common (left) to least (right); only the five most common are shown. Percentages reflect the fraction of times the adjective was paired with the noun.	117

4.5	ANPs for negative sentences generated by <i>SentiCap</i> . Nouns are ordered from most common to least, with only the ten most common shown. Paired adjectives are ordered most common (left) to least (right); only the five most common are shown. Percentages reflect the fraction of times the adjective was paired with the noun.	118
4.6	The mean number of each POS class per sentence for the positive sentiment generated sentences. Note <i>CNN+RNN</i> generates MSCOCO style captions all other methods generate positive sentiment.	119
4.7	The mean number of each POS class per sentence for the negative sentiment generated sentences. Note <i>CNN+RNN</i> generates MSCOCO style captions all other methods generate negative sentiment.	119
5.1	Results for the end-to-end sentence simplification task. Our complete model is <i>S4+gv+bce</i> . Section 5.4.1 details the metrics. Values in bold are closest to the ground-truth. For BLEU or Rouge the largest value is closest to the ground-truth, for Flesch or Average Words the closest has the smallest delta from a Flesch of 74.69 or a Average Words of 15.72.	138
5.2	Simplification examples from the <i>S4+gv+bce</i> model. Bold-face highlights changes from the original complex sentence to the simplified sentence. Insertions and replacements are bold-face in the simplified sentence, while deletions are bold-face in the complex sentence. <i>Orig</i> is the original complex sentence, <i>Simp</i> is the output of the <i>S4+gv+bce</i> model, and <i>GT</i> is the ground truth simplified sentence.	141
5.3	BLEU and Rouge scores when an oracle word simplifier is used: when the correct alignment is made the chosen word is guaranteed to be correct. This measures the performance of the attention layer alone. . .	141
5.4	Confusion matrix for choosing to change or copy a word. The rows are the actions chosen by the <i>S4+gv+bce</i> model when fed ground-truth alignments. The columns are the ground truth actions.	142
6.1	The most common frames in the MSCOCO (596K training captions) and Romantic Novels Dataset (578K training sentences) with their frequency count and most common verbs.	151
6.2	Evaluating caption descriptiveness on MSCOCO dataset. For metrics see Sec. 6.4.4, for approaches see Sec. 6.4.2.	165
6.3	Evaluating styled captions with automated metrics. For <i>SPICE</i> and <i>CLF</i> larger is better, for <i>LM</i> & <i>GRULM</i> smaller is better. For metrics and baselines see Sec. 6.4.4 and Sec. 6.4.2.	165

6.4	χ^2 tests on method pairs for human story judgements . We combine counts for “unrelated” with “purely descriptive”, while “story” is kept as its own class. Those marked with a * indicate rejection of the null hypothesis (H0: the two methods give the same multinomial distribution of scores) at p-value of 0.005 – this is p-value of 0.05 with bonferroni correction of 10 to account for multiple tests.	169
6.5	χ^2 tests on method pairs for human descriptiveness judgements . We combine counts for “clear and accurate” with “only a few mistakes”, and “some correct words” with “unrelated”. Those marked with a * indicate rejection of the null hypothesis (H0: the two methods give the same multinomial distribution of scores) at p-value of 0.005 – this is p-value of 0.05 with bonferroni correction of 10 to account for multiple tests.	169
6.6	Precision (BLEU-1) and recall (ROUGE-1) in our semantic term space. . . .	171
6.7	Style attribute statistics based on 4000 random ground-truth sentences for MSCOCO and romance styles and 4000 test captions generated by the descriptive only model (CNN+RNN-coco) and our <i>SemStyle</i> model. We measure the percentage of sentences or captions with present tense root verbs, past tense root verbs, and first person pronouns. We also count the number of unique verbs used in the sampled sentences. . . .	173
6.8	The most common words per part-of-speech category in the two ground-truth datasets and in the sentences generated by the descriptive model (CNN+RNN-coco) and <i>SemStyle</i> . For each word we display the relative frequency of that word in the POS category – represented as a percentage.	174

Introduction

An artificial agent needs to be able to perceive the world state and take action as a result. One particularly useful aim is to develop an agent that perceives the physical world and takes action through communication with human collaborators. A powerful tool for perceiving the physical world is sight. The study of computer vision aims to provide sight to artificial agents, enabling them to understand complex visual scenes. As a core topic in artificial intelligence and machine learning it has been the focus of extensive research, but is far from solved, with humans still outperforming artificial vision systems in most tasks. Communication between humans is primarily through language. Designing an agent that can communicate via language is an important goal for human-agent interaction and for building agents that can learn from the vast repositories of human knowledge. With these aims natural language processing is a core topic in artificial intelligence and machine learning. Like computer vision, natural language processing has been the focus of extensive research, but remains an open problem. This thesis seeks to connect two core topics in machine intelligence: vision and language. Although several topics exist at the intersection I focus on automatic image captioning: generating natural language descriptions of image content. Automatic captioning involves both the image understanding problem from computer vision and the natural language generation problem from natural language processing. To improve communication I endeavour to add an extra layer to automatic captioning in the form of linguistic style. Stylistic variations in language have a range of useful applications, such as: reaching a broad audience, reducing misinformation, and engaging viewers. With these applications in mind I develop and evaluate novel methods capable of generating stylised captions for natural images.

Image captions typically consist of a range of concepts visually identified in the image such as, objects, actions, attributes and scenes. To form natural language, these concepts need to be woven into grammatically correct sentences e.g. *"The dog runs through the grass."*. Automatic image captioning has already be used to improve ac-

cessibility for vision impaired users [Wu et al., 2017], and has promising applications to news reporting [Feng and Lapata, 2010] and foreign language learning [Winke et al., 2013].

Linguistic style can be loosely defined as *how* something is written rather than *what* is written. Said differently, when changing style we change the textual realisation without changing the semantics. However, completely preserving semantics across different writing styles without adding or removing any content is an impossible task since any suitably rich style should convey information that can shape a readers interpretation of the content. Hence, I do not strictly apply this definition. In the case of image captions, the key goal is generating captions in the target style while retaining the semantics necessary for visual relevance. This is an exceedingly broad view of style compared to its formal usage within linguistics [Simpson, 2004]. Other authors have taken similarly broad views of style [Shen et al., 2017; Fu et al., 2018; Prabhumoye et al., 2018] as way of making progress in the task of generating styled text. Moreover, drawing a line between changes in style as apposed to content is a challenging problem even within linguistics, for example to achieve a particular stylistic goal it might be necessary to change the content of a sentence. Could such a change still be considered stylistic? It seems that there are in-fact two mutually dependent parts to such a change, one part is a content change while the other part is a style change. In such cases the task is to trade-off the change in style with the change in content. In this thesis I will, among other things, cover the tasks of generating language with a particulate sentiment, and making sentences simpler. In these tasks it is hard to argue that a change in style is not needed; however, the extent to which each task involves changing the content depends on how you define the border between style and content. I will avoid making a fine grained distinction, and instead consider these as style tasks, such that changes made in pursuit of the desired language are stylistic. For this thesis it is most important to ensure each of these tasks are well-defined rather than make a fine grained distinction.

To fully realise the benefits of automatic image captioning, a degree of flexibility should be built into these systems. The same description may not be appropriate or useful for all audiences, for example, it is reasonable to describe an image differently to a child, rather than a domain expert. Moreover, the caption style used in personal social media posts may not be appropriate for news articles. Online product reviews matching the writing style of the target market have a greater influence on purchasing decisions than those which fail to do so [Ludwig et al., 2013]. Linguistic style is also known to reflect personality [Pennebaker and King, 1999; Oberlander and Gill, 2006] and foster social interactions [Danescu-Niculescu-Mizil et al., 2011; Doyle et al., 2016]. By introducing style into automatic image captions I hope to emulate skilled

human authors who adapt their style to the audience as a way of easing communication, or forming a connection with a particular social group [West and Turner, 2010; Danescu-Niculescu-Mizil et al., 2011; Doyle et al., 2016]. This thesis is the first to explore linguistic style for automatic image captioning.

1.1 Main Research Challenges

The primary goal of this thesis is the development of methods for describing image contents in appropriately styled natural language. This is a complex task that I break down into three major sub components: style and content representation, generative models for styled captions, and methods for overcoming data scarcity. These components align with three core parts of any machine learning solution: representation, modelling, and training strategy.

1.1.1 Style and Content Representation

Representing both style and content is an important part of stylised caption generation. An ideal representation would separate text into independent content and style dimensions, while accurately capturing both content and style. Separation enables independent control over content and style and also allows new styles to be learnt from text other than image captions. Accurate capture of content enables the generation of visually relevant captions, while accurate capture of style enables the generation of appropriately styled captions. Constructing such a representation is an open problem, attracting interest from both psychology [Bebout, 1993; Coltheart, 1981] and machine learning [Tausczik and Pennebaker, 2010; Gan et al., 2017a]. However, there is no consensus. This is in part because the definition of style is itself contentious [Bebout, 1993; Simpson, 2004] and because compactly representing sentence content is also a challenging problem [Le and Mikolov, 2014; Kiros et al., 2015; Conneau et al., 2017; Cer et al., 2018] with a degree of domain dependence.

Developing a representation for style requires us to define what aspects of style and content to encode and how they are separated. The style can be defined explicitly by a discreet set of attributes such as: sentiment, formality, and complexity. As these attributes only weakly describe the style it is typically necessary to use them as a guide to data collection so as to learn the style representation from labelled data. Alternatively, style can be defined implicitly by a text corpus of a consistent style (eg a particular author or genre). The separation between style and content can be defined via manual annotation of sub-components, such as n-grams [Tausczik and Pennebaker, 2010], or learnt using ground-truth style labels of various granu-

larities [Tenenbaum and Freeman, 2000; Popa et al., 2009; Gan et al., 2017a]. In this thesis I explore different representations of content and style and different methods for extracting these representations. The focus is on representations useful for generating styled image captions, rather than generic representations. I frequently define narrow style attributes to be considered, and then proceed to learn the realisation of this style from data. The final style representation is therefore encoded in the model parameters. When defining image-caption content, I fall back on visual concept detectors and image-caption datasets written in a purely descriptive style.

By considering style as the variation in textual realisation when content is controlled, Chapter 3 explores the distribution of names used to describe visual concepts within their context. This provides insights into caption style variability and a method for choosing names that are more natural for a given context. In Chapter 6, style is defined by a binary sentiment attribute. Generic sentiment dictionaries provide sentiment labels for adjective noun pairs. To adapt these sentiment labels to entire image captions I develop a crowd-sourced dataset construction task. This new dataset of sentiment captions and an existing set of descriptive captions is used to train a neural network model with separate components for sentiment and descriptive words, explicitly separating style and content. In Chapter 6 I propose a procedure for approximately separating image-caption content from its styled realisation. Using these rules, I train a sequence-to-sequence model that encapsulates different text styles. In this case the style is most appropriately described as the distribution over sentences for a given content representation.

1.1.2 Generative Models for Styled Captions

Generated captions should reflect both content and style attributes. Captions that lack visual relevance are not useful image descriptions, while those that lack the target style forgo the benefits of written style. Models for stylistic captioning must trade-off between – the not entirely separable attributes – style and content. Modelling this trade-off is one of the more challenging aspects of styled caption generation. Limited styled captions means learning the whole model on a single dataset is often impractical, so for example this trade-off might be an explicit weighting between specialised style and content components. The specific form of the generative model can limit the types of style that can be realised in the output sentence. For example, style can be conveyed through both word choice and sentence structure, but developing models capable of both is a challenge – whether both are necessary depends on the style. Modelling styled captions also has many of the same challenges as modelling descriptive captions: detecting objects, actions and attributes in

the image; selecting what to describe in the caption; and generating fluent natural language. Recent advances in object detection [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016] and natural language generation [Bengio et al., 2003; Mikolov et al., 2010; Graves, 2013] have made steps towards solving these descriptive captioning challenges [Donahue et al., 2015; Karpathy et al., 2014a; Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015b]. I build on these advances and introduce several novel styled captioning models with varying degrees of style control.

In Chapter 4 I develop a generative model for styled image captions with dynamic switching between specialised components for style and descriptive words. The component controlling this switching explicitly balances style and content in the output, with the optimal trade-off defined by a dataset. This generative model is most suitable for styles expressed by small atomic changes to descriptive captions, such as word replacement or insertion. Both positive and negative sentiment styles can be generated by this model; however, they must be trained separately, and the model is not capable of switching between them at test time. Chapter 5 tackles more nuanced style generation with a sequence-to-sequence model for simplifying news article sentences. Simplification – where sentence complexity is reduced to aid those with limited reading ability – requires more complex style mimicry, including word replacement, sub-string removal and sentence splitting. A custom objective helps to balance semantic preservation – here implemented with input to output word copying – and simplification. Using news articles provides the added challenge of modelling a large content domain: methods for dealing with this are proposed. Chapter 6 focuses on generating captions in a style which is learnt from a large text dataset. The model separates the generation of style and content using a two stage pipeline, where a purely descriptive component identifies image content and selects what to describe, and a second component realises these concepts in the desired style. The form of the second component and the training procedure ensure a strong connection between the image content and the output caption. Part of the input to this second stage is a style flag, allowing different styles to be generated from the same model.

1.1.3 Data Scarcity

A key goal of this thesis is reducing the burden of data collection for styled image-caption generation. Existing methods for image captioning learn visual semantics and language construction simultaneously [Donahue et al., 2015; Karpathy et al., 2014a; Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015b], necessitating a large training set with coverage of both aspects. Even the collection of styled image-caption

datasets is difficult because of the high level linguistic skills required by annotators. In applications where individualised style matching is important, such as aiding communication or encouraging a desired emotional response, it may not be possible to collect a comprehensive dataset in advance. Moreover, there are a large number of possible linguistic styles [Pavlick and Nenkova, 2015; Fidler and Goldberg, 2017; Xu, 2017], defined by attributes such as sentiment, simplicity, formality, voice, genre or author. This makes collecting image-caption datasets for every style of interest impractical, especially since even for a single style this is an expensive task. I demonstrate methods to ameliorate the data scarcity problem by using: transfer learning, separate components for style and content, and external data sources. I also explore dataset construction techniques that are less demanding of annotators.

In Chapter 3 I learn how to choose object names using a web-scale image-caption dataset collected from social media. This technique uses raw data and requires no additional manual annotation. Although this model does not generate full captions, it demonstrates that social media data mining is a promising direction for uncovering captioning styles used in the wild. Chapter 4 exploits existing descriptive image-caption datasets with a method for explicitly merging a descriptive model with a style model fine-tuned on a small number of styled captions. The dataset collection mixes existing descriptive captions with pre-defined word level style labels in a rewriting task suitable for crowd-sourcing where annotators exhibit highly variable linguistic skill. Chapter 5 presents a model for sentence re-writing, which reduces data requirements by exploiting pre-trained word embeddings and the similarities between the input-output sentences. Chapter 6 works towards fully separate training for style and content components in order to eliminate the dependence on training captions in the target style. The resulting model generates convincing, visually grounded, styled captions without ever training on styled captions.

1.2 Thesis Outline

Image caption generation, and in particular styled caption generation, is a complex task consisting of a number of moving parts and possible variations. I start simply and then build up to more complex style generation. Chapter 3 is concerned with generating name variations for visual concepts, but does not cover sentence generation. The following chapter (Chapter 4) tackles full sentence caption generation for images but is tuned for styles which can be expressed as small rewrites of descriptive captions. Concerned only with text, Chapter 5 explores methods for rewriting sentences so that they conform to linguistic style objectives. The rewrites considered go beyond word replacement, including sentence re-ordering or splitting as well as

sub-sequence removal. Finally, Chapter 6 brings together the previous ideas and developments to demonstrate styled caption generation with a high degree of linguistic flexibility.

In Chapter 3 I explore how people name visual objects in image-captions. Naming is key to how we interpret the physical world [Lakoff, 1987], so by understanding and predicting naming choices I lay the ground work for accessible and attractive image captions. While previous work [Ordonez et al., 2013] has looked at naming with large image-caption collections, I take this a step further and explore the affect of visual context and the consistency of naming choices. This verifies visual context as an important factor in naming – a relationship previously observed through small-scale controlled experiments [Rosch, 1999; Chaigneau et al., 2009]. Moreover, I train a contextual naming model on a web-scale image corpus capable of naming a larger range of objects than previous systems, while scaling easily to even more objects. As a brief extension, I consider the case of animal naming in the highly structured Linnaean hierarchy. This domain allows analysis of naming specificity and consistency across sub-trees at scale. This chapter explores naming within image captions, while the generation of full sentence captions is covered in the following chapters.

Chapter 4 develops ideas for modelling styled image-captions when limited styled data is available. The styles considered are positive and negative sentiment, which is interesting in its own right because sentiment can drive decision making [Lerner et al., 2015; Ludwig et al., 2013]. A small sentiment caption dataset was collected specifically for this task and used alongside a large existing descriptive caption dataset. The new dataset collection technique, involving guided sentence rewriting, could also be adapted to the collection of similar styled image-caption datasets in the future. Human validation of the sentiment dataset reveals that a caption’s sentiment polarity is not strongly constrained by the image, so we have some freedom to choose the desired polarity when generating. The captioning model, named SentiCap, employs two recurrent neural network streams joined by a learnt switching network. One recurrent network is responsible for generic descriptive words, while the other is tuned for words expressing a strong sentiment. SentiCap demonstrates the ability to generate captions that are both semantically relevant and express either positive or negative sentiment. The SentiCap modelling techniques apply to styles that can be expressed with small localised edits to descriptive captions, such as word replacement, insertion, and deletion. More complex stylistic changes are beyond the scope of this chapter, but are explored in subsequent chapters.

Chapter 5 eschews caption generation to focus on developing language modelling techniques for styled sentences. I consider text simplification: rewriting existing sentences to ease understanding. The training dataset consists of parallel complex and

simplified sentences from news articles. The model, called S4 (Sequence-to-Sequence for Sentence Simplification), is a sequence-to-sequence recurrent neural network with attention and word copying connecting the input and output sentences. S4 is capable of complex edits, including word replacements, structural changes and sentence splitting. As data scarcity is a problem, I present some broadly applicable techniques for reducing the data requirements of mono-lingual sentence-to-sentence re-writing tasks. In particular, I use a mix of fixed pre-trained and learnable word embeddings to overcome dataset limitations, particularly in regard to out of vocabulary words. The loss function designed to encourage word copying helps to reduce dataset requirements by focusing on sentence changes and therefore using model capacity more efficiently.

Chapter 6 tackles the problem of generating styled image-captions when the style is defined by a large text dataset not paired with images. In this case, I use a collection of romance novels to define the style. The model, called SemStyle, is a two stage recurrent neural network pipeline, where the first stage handles object recognition and content planning, while the second stage handles text realisation to meet both content and style goals. The second stage allows a high degree of linguistic flexibility by using sequence-to-sequence modelling inspired by the previous chapter. A shared intermediate representation is used for passing information between the two stages. I found a discreet token representation, defined by linguistic heuristics, to be the most appropriate representation, when compared with multi-modal vector space representations. Using this intermediate representation allows style to be learnt separately from image semantics. The complete SemStyle model shows competitive results in human and automatic evaluations with respect to existing descriptive metrics and two new automated metrics for style. The other advantages of SemStyle are: reduced dataset requirements, simultaneous style expression with visual relevance, and the ability to learn new styles from text without additional guidance.

1.3 Main Contributions

The main contributions of Chapter 3 are:

- The large-scale verification of visual context as an important factor in object naming.
- A new method for predicting context-dependent names taking into account visual and linguistic information, which shows substantial improvement on the image-to-word task.

-
- Evaluations of naming on a dataset two orders of magnitude larger than prior work [Ordonez et al., 2013].
 - An analysis of the specificity of animal naming.

The main contributions of Chapter 4 are:

- A model for generating visually relevant positive and negative sentiment captions.
- A learnable switching component that reduces training data requirements by balancing language models for style and visual relevance.
- Techniques for crowd-sourcing stylistic captions from annotators with limited linguistic knowledge.
- An exploration of the suitability of different sentiment polarities to images.

The main contributions of Chapter 5 are:

- A sequence-to-sequence model for sentence simplification.
- A technique for dealing with sparse word usage during training which employs both pre-trained and learned word embeddings.
- A novel loss function to encourage input to output word-copying where appropriate – especially effective when paired with generation-time copying.

The main contributions of Chapter 6 are:

- A system for generating visually relevant image captions in a target style without paired training data.
- A concise semantic term representation for image and language semantics.
- A comparison of the semantic term representation with multi-modal vector space representations.
- Competitive results in human and automatic evaluations with existing, and two novel automated metrics for style.

Related Work

The generation of stylised image captions involves three key areas: image understanding, natural language generation, and style modelling. Image understanding in this case refers to detecting key image concepts, such as object, actions and attributes. Recent advances [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016] in the image understanding area have enabled more accurate image caption generation. Natural language generation involves constructing semantically relevant and grammatically correct sentences, such as image captions. Neural network language models [Bengio et al., 2003; Mikolov et al., 2010; Graves, 2013] have become a powerful tool for language generation, allowing both long sequence memory and joint training with image understanding components. Style modelling consists of defining style, separating style from content, and generating text in a given style. Style is loosely defined within the literature, so to ease its application in an algorithmic setting a more specific definition is required. For example style may be defined in terms of a document collection or a specific attribute such as sentiment or simplicity. Separating content from style is also important because it ensures semantics and style can be specified separately. Generating styled text is a variation on natural language generation, which has additional stylistic objectives.

This chapter reviews the three key areas for stylised image captioning: image understanding, natural language generation, and style modelling. I also cover descriptive caption generation, which is at the intersection of image understanding and natural language generation, and styled text generation, which is at the intersection of natural language generation and style modelling. These areas form the basis of styled caption generation techniques developed in later chapters. Finally, I review the limited literature that focuses on stylised caption generation.

General image captioning concepts are covered, followed by a discussion of linguistic style and its applications to text generation and image captioning. The literature reviewed here provides the background for understanding stylistic image captioning techniques. Related work sections in each chapter cover literature specifically

relevant to that chapter. First, in Section 2.1, I review object recognition, which is the form of image understanding used in this thesis. The main focus is Convolutional Neural Networks (CNNs) which form the backbone of most modern approaches to object detection in natural images. Language generation is reviewed in Section 2.2, with a focus on Recurrent Neural Networks (RNNs): a flexible tool for generating language. Putting these pieces together are the caption generation approaches reviewed in Section 2.3, after which, I shift focus to style, starting with definitions of style and its importance to human interactions in Section 2.5. This leads into a review of style and content separation in Section 2.5. Recent approaches for generating stylised text are reviewed in Section 2.6. Techniques for stylised caption generation are reviewed in Section 2.8.

2.1 Object Recognition

One of the first steps in image caption generation is identifying image contents, this is where object recognition is relevant. Many of the models presented in this thesis rely on object recognition to identify image contents. This section provides background information on object recognition, and highlights some of the challenging aspects of the problem and their affect on real world performance.

Identifying the contents of an image is a core computer vision problem. There are many instances of this problem, such as: image classification, where the whole image is given a single label; object detection, where multiple objects are detected per image; object detection and localisation, where location in the image must also be identified; and segmentation, where every pixel in the image is classified. In each of these cases, the goal is to automatically label an image with its contents. This review focuses on image classification and object detection as they are of key importance to the methods developed in this thesis. Doing this simplifies the discussion by avoiding the complex output smoothing approaches necessary for high quality image segmentation. Since object detection is more general than image classification, and many image classification techniques can be adapted for object detection, I adopt the convention of referring to both as object detection except when the distinction is pertinent to the discussion.

Object detection with and without localisation has many practical applications including face detection for digital cameras [Viola and Jones, 2001], digit recognition for postcodes [LeCun et al., 1998], pedestrian detection [Dalal and Triggs, 2005] and restaurant food identification for diet tracking [Bossard et al., 2014]. Object detection approximately meets or exceeds human performance in tasks such as face verification [Lu and Tang, 2015] and handwritten digit recognition [Ciregan et al., 2012].

Image classification accuracy for natural images has exceeded human level performance [He et al., 2015], though this only applies in the special case of top-5 selection on 1000 fine-grained classes. Most human errors were from difficulty differentiating similar classes (eg different dog breeds) or because of a lack of class knowledge; in general, humans are still better at object recognition.

Although humans take for granted the ability to visually recognise, it is a difficult task to solve algorithmically. The core problem is high intra-class appearance variability: instances of a concept may be diverse in appearance, for example an office chair looks different to a dining room chair. Even identical objects exhibit appearance variability because of: lighting conditions, imaging noise, translation, perspective, rotation, occlusion and object deformation [Pinto et al., 2008; Russell and Norvig, 2010]. Practical object recognition algorithms must model appearance in a way that is robust to these factors, and specific enough to avoid misclassification. Computational feasibility is clearly also necessary.

Many methods have been proposed for object detection. Early techniques relied on hand designed rules or model matching [Roberts, 1963]. These techniques were difficult and costly to develop, and offered poor generalisation to appearance variations. As a result, modern object detection algorithms are based on statistical machine learning, allowing appearance variation to be learnt from data rather than hard-coded into the system. Notable approaches include eigenfaces [Turk and Pentland, 1991], harr cascades [Viola and Jones, 2001], support vector machines [Cortes and Vapnik, 1995; Phillips, 1999; Dalal and Triggs, 2005], neural networks [LeCun et al., 1998] and deformable parts models [Felzenszwalb et al., 2010]. Until very recently, state-of-the-art methods worked by applying a classifier to extracted features, rather than raw pixels. The current state-of-the-art methods are based on Convolutional Neural Networks (CNNs) that when provided with a large enough training set, may learn features directly from data [Krizhevsky et al., 2012].

Designing features manually for the vision domain is challenging as ideally they are invariant to appearance variability, without sacrificing discriminability. Well-designed features incorporate some prior knowledge of the task, allowing simple classifiers to be used, and giving more accurate results with less training data. Common feature choices include: colour histograms [Swain and Ballard, 1991], edges [Canny, 1986], histograms of orientation gradients (HOG) [Dalal and Triggs, 2005] and scale invariant feature transform (SIFT) [Lowe, 2004] features.

CNNs have recently become the state-of-the-art for object detection and a number of other related vision tasks. Unlike previous approaches, which required careful feature design, CNNs work directly with raw pixels, building up multiple layers of features [Zeiler and Fergus, 2014] during training – this is the essence of deep learn-

ing. The main factors contributing to the success of CNNs are: a large increase in the volume of training data [Russakovsky et al., 2015], cheap and high performance GPU hardware [Oh and Jung, 2004; Raina et al., 2009], and improvements in deep neural network training [Krizhevsky et al., 2012; Srivastava et al., 2014; Kingma and Ba, 2015; He et al., 2016; Nair and Hinton, 2010; Glorot and Bengio, 2010]. More recent advances have been spurred on by flexible high performance automatic differentiation libraries [Bastien et al., 2012; Abadi et al., 2016; Collobert et al., 2011], making experimentation with new architectures more accessible and reducing iteration time. Higher level libraries [Donahue et al., 2014; Chollet and Others, 2015; Jia et al., 2014; Seide and Agarwal, 2016] have allowed CNNs to spread into other fields such as astronomy [Dieleman et al., 2015] and biology [Ciresan et al., 2012].

2.1.1 Neural Networks for Classification

Neural network models are used extensively in this thesis for both object detection and language generation. This section provides general background on neural networks, including their structure and training.

An artificial neural network [Rosenblatt, 1958] is constructed by composing non-linear functions that act on the weighted sum of their input. This basic building block is sometimes called a neuron, because of a crude analogy to the functioning of neurons in the brain. This analogy is mostly historical, and much more powerful models exist for simulating neurons [Hodgkin and Huxley, 1952]. We can formally define a neuron as a function applying a non-linearity $\sigma(\cdot)$ (called the activation function) to a weighted vector of activations \mathbf{a}_{in} and producing a scalar output activation a_{out} (some authors include an additive bias b , which we omit because it is equivalent to appending an extra unit activation to the input of each neuron),

$$a_{out} = \sigma\left(\sum_{i=1}^N w_{ji} a_{in,i}\right) \quad (2.1)$$

Typically neurons are arranged in layers, with the output of proceeding layers used as input, forming what is known as a feed-forward network. In the modern neural network literature, the term layer can mean anything from a complex chain of operations to something as simple as an activation function; however, in general they take the tensor output of one or more previous layers, apply a function, and return a new output tensor. We can compactly define a layer of neurons indexed by j with weight matrix W_j by rewriting Equation 2.1 using vector notation:

$$\mathbf{a}_j = \sigma(W_j \mathbf{a}_{j-1}) \quad (2.2)$$

The input to a neural network of depth D is \mathbf{a}_1 , while the output is the activations of the final layer \mathbf{a}_D .

For intermediate layers, the non-linearity $\sigma(\cdot)$ was traditionally the sigmoid function $\sigma(x) = \frac{1}{1+\exp -x}$ or $\sigma(x) = \tanh(x)$; however, the rectified linear unit (ReLU) $\sigma(x) = \max(0, x)$ is now the dominant choice. ReLU tends to converge faster [Goodfellow et al., 2016; Nair and Hinton, 2010], because its gradients remain linear for large x , in comparison, for large x ; the gradient of the sigmoid approaches 0. For the last layer, the activation function tends to be more problem specific. For example in multi-class classification the softmax function $\sigma(x) = \frac{\exp x}{\sum_{k=1}^K \exp x_k}$ is the standard choice.

For multi-class classification over C classes, a common loss function is categorical cross-entropy between the final layer activations a_D and the ground truth label \hat{y} :

$$\mathcal{L} = - \sum_{i=1}^C \mathbb{I}[i = \hat{y}] \log a_{D,i} \quad (2.3)$$

With $\mathbb{I}[\cdot]$ the indicator function. In practice we can reduce this to:

$$\mathcal{L} = - \log a_{D,\hat{y}} \quad (2.4)$$

Minimising the categorical cross-entropy is equivalent to maximising the likelihood of the data under the categorical distribution parameterised by the neural network [Murphy, 2012, Sec.28.4.2]. The main advantage of this loss function is a numerical stable derivative that is robust to saturation when composed with the softmax activation [Goodfellow et al., 2016, Sec.6.2.2.3]. Other loss functions such as least squares do not have this property. The derivative of the cross-entropy loss function \mathcal{L} with respect to the softmax activation $a_{D,i}$ is:

$$\frac{\partial \mathcal{L}}{\partial a_{D,i}} = a_{D,i} - \mathbb{I}[i = \hat{y}] \quad (2.5)$$

Note that this form assumes $a_{D,i}$ uses a softmax activation.

The parameters of a neural network are typically learnt from data using back-propagation [Rumelhart et al., 1986] and a gradient descent optimiser. Other learning algorithms have been suggested [Bengio and Frasconi, 1994] but back-propagation is the dominant method because it: is easy to apply, scales well, and generally produces good solutions. Back-propagation calculates the adjustments to networks weights W_j by taking the partial derivatives of the loss function with respect to each of the weights in the network. The chain rule is used to calculate the gradients of parameters in earlier layers (layers closer to the input) given later layers. This allows for efficient computation and gives rise to the name back-propagation. A gradient descent optimiser such as Stochastic Gradient Descent (SGD) [Widrow and Hoff, 1960; Bottou, 1998], RMSProp [Tieleman and Hinton, 2012] or Adam [Kingma and Ba, 2015] iteratively updates the weights using the calculated gradients. Within each iteration, the gradients are approximated using a small random set of examples called

a mini-batch. The noise in this approximation helps to find flat local minima [Keskar et al., 2017] which tend to generalise well [Smith and Le, 2018].

We formalise back-propagation by roughly following Nielsen [2015]. The error vector δ_j for each layer j is defined using the weighted activations $z_j = W_j a_{j-1}$, as:

$$\delta_j = \nabla_{z_j} \mathcal{L} \quad (2.6)$$

With $\nabla_{z_j} = (\frac{\partial}{\partial z_{j,0}}, \dots, \frac{\partial}{\partial z_{j,H}})$ for a layer consisting of H neurons. Using the chain rule the error of the output layer δ_D is:

$$\delta_D = \nabla_{a_D} \mathcal{L} \odot \sigma'(z_D) \quad (2.7)$$

With \odot , point-wise multiplication and a_D the activation for the last layer defined by Equation 2.2. In the case of a softmax non-linearity $\nabla_{a_D} \mathcal{L}$ is computed by Equation 2.5. The errors for earlier layers can be calculated using the errors from later layers as:

$$\delta_{j-1} = (W_j^T \delta_j) \odot \sigma'(z_{j-1}) \quad (2.8)$$

Using Equation 2.8, we step backwards through the network calculating the errors in a layer-wise fashion. The derivatives of the loss with respect to the weight are calculated from the errors and stored activations:

$$\nabla_{W_j} \mathcal{L} = a_{j-1} \delta_j^T \quad (2.9)$$

The computed derivatives can then be used in a gradient descent optimiser such as SGD. In the above, we assumed a particularly common form for a_j (Equation 2.2). Many neural network architectures use different forms requiring changes to Equations 2.8 & 2.9.

2.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the primary object detection model used in this thesis. This section provides background on CNNs, briefly touching on the range of architectures that have been used in this thesis. Included are some practical considerations and tricks for designing CNNs that are also applicable to other types of deep network. However, the design of CNNs is not a focus of this thesis, so the discussion is kept brief.

CNNs [LeCun et al., 1998] are a type of artificial neural network using convolution layers; they have been applied in different domains such as text [Santos and Gatti, 2014; Zhang et al., 2015], audio [Abdel-Hamid et al., 2012, 2014] and video [Karpathy et al., 2014b; Ng et al., 2015], although this review focuses on object detection applied to static images. In this case, the CNN takes raw pixels (normally

with mean subtraction and scaling applied) as input, and outputs a categorical distribution over object classes. Techniques and ideas that apply to neural networks – Section 2.1.1 – also apply in the special case of CNNs; however, we must also add a number of special purpose layers such as convolution and pooling layers.

A typical CNN for classification uses stacked convolutional layers, followed by a number of densely connected layers, and finally a softmax activation. The convolutional layers are themselves composed of a convolution stage, a non-linearity, and a pooling stage. The non-linearity is one of the activations reviewed in Section 2.1.1, with ReLU the most common. Convolution and pooling are described below.

Convolution is the core of the CNN. We can define a convolution of a 2-d image I and 2-d weight matrix as,

$$S(i, j) = \sum_{m=0}^M \sum_{n=0}^N I(i - m, j - n) W(m, n) \quad (2.10)$$

where i and j are the image row and column. Here the size of the convolution is $M \times N$, typically this is $K \times K$ for $K = 3, 5, 7$. The boundary of the image is often padded with zeros so the convolution gives an output of the same size (a *same* convolution), although other options are possible (*valid* and *full*). We can extend convolution to colour images with 3-dimensions by taking the dot product of vectors $I(i - m, j - n)$ and $W(m, n)$. In practice $I(i - m, j - n)$ is a matrix because of the mini-batch dimension, and W is a matrix because we apply multiple convolutions at once. In effect the image batch is a 4-d tensor $batch_size \times rows \times columns \times colour_channels$ that our set of convolutions maps into another 4-d tensor of $batch_size \times rows \times columns \times feature_channels$. Another way of understanding convolution in CNNs is the application of a small fully connected layer to every offset in the image. Since weights are shared across all offsets, the number of learn-able parameters can be much lower than a single fully connected layer for the entire image. Moreover, the features are both localised and invariant to translations in the original image.

Pooling replaces the activations of a region with a summary statistic such as max, mean or l2-norm [Goodfellow et al., 2016]. In CNNs on images, it is common for the region to be 2×2 over spatial dimensions, with a stride of 2 resulting in an output of half size in both dimensions – the feature dimension is not modified. Max pooling is the most common variant for CNNs because it has produced empirically better results than alternatives [Scherer et al., 2010]. Pooling introduces local feature translation invariance, which helps the network generalise to small deformations in appearance [Boureau et al., 2010]. Although variable pooling sizes would allow the network to be applied over variable sized input images, in practice images are resized via bilinear or bi-cubic interpolation prior to being fed into the network.

A large number of different CNN architectures have been published, those of major historical importance include: LeNet [LeCun et al., 1998] which was the first CNN and AlexNet [Krizhevsky et al., 2012] which won the ILSVRC challenge in 2012 and kicked off a frenzy of CNN architectures for image classification. More recently VGGNet [Simonyan and Zisserman, 2015], showed that the depth of the network is an important factor in performance. They produced models of 16 and 19 layers using only 3×3 convolutions and 2×2 pooling. This extremely elegant architecture has been used in a number of different publications [Ren et al., 2015; Fang et al., 2015; Bernardi et al., 2016], becoming the standard CNN architecture for a time.

Currently the state-of-the-art is occupied by architectures based on ResNet [He et al., 2016] and Inception [Szegedy et al., 2015, 2016], though these architectures are not mutually exclusive [Szegedy et al., 2017; Xie et al., 2017]. ResNet uses residual connections: an identity transform layer skips multiple learnt transform layers before the features are added back together. The learnt transformations in the residual block can be seen as modelling the residual of the underlying mapping (the ideal mapping) and the input features. Residual connections combined with batch normalisation [Ioffe and Szegedy, 2015] to mitigate the problems of vanishing and exploding gradients made it possible to train networks with 152 layers – far exceeding the depth of previous approaches. Inception architectures make use of the inception layer composed of parallel 1×1 , 3×3 and 5×5 convolutions also in parallel with 3×3 max pooling. In practice, 1×1 convolutions are applied to reduce the size of the feature dimensions. This allows inception layers to exploit multiple scales while keeping the number of trainable parameters to a minimum. Employing an ensemble of Inception and ResNet models combined with a new channel weighting layer, Hu et al. [2017a] won the ILSVRC 2017 classification challenge with their SENet architecture. Channels are weighted by an MLP with two hidden layers applied to the average channel value across all spatial locations: the MLP outputs one weight for each channel.

Regularization is necessary for training CNNs except in rare cases where there is a wealth of data – typically tens or hundreds of millions. The most common form of regularization is early-stopping: using a validation set to assess generalisation error during training, and terminating when this starts to increase. This is easy to implement and comes with a computational advantage. Another popular approach is dropout [Srivastava et al., 2014], where a randomly selected fraction of layer activations are set to zero on each training run. This is also easy to implement, has minimal impact on performance and is an effective regularization technique. It is most commonly applied to fully connected layers, and rarely in convolution layers. Dropout can be viewed as a strategy for temporarily removing nodes from the neu-

ral network – forming a similar, but different network on each training iteration. Using this interpretation, Srivastava et al. [2014] argue dropout is able to approximately combine exponentially many neural networks. Other dropout variants have been proposed, such as dropping out connections [Wan et al., 2013], or entire convolution features [Tompson et al., 2015]. Weight decay regularisation via L2 and L1 norm are sometimes applied, but it is relatively uncommon in CNNs. Batch normalisation [Ioffe and Szegedy, 2015] can also be a regularizer because it modifies activations with statistics from the randomly selected examples in each mini-batch. With batch normalisation, other regularisation techniques such as dropout may not be necessary.

Neural networks require some initial choice of parameters, called weight initialisation. Choosing poorly can substantially slow down training, cause convergence to a high loss local minima, or prevent convergence altogether [Glorot and Bengio, 2010; Simonyan and Zisserman, 2015]. Most important is ensuring the initial parameters break symmetry: that is, neurons with identical inputs should be initialised with different input weights to avoid redundancy [Goodfellow et al., 2016, Sec.8.4]. A simple and quite popular initialisation approach is to sample weights from a uniform or gaussian distribution with scale carefully chosen to avoid exploding or vanishing gradients. Theoretically, grounded heuristics [Glorot and Bengio, 2010; He et al., 2015] exist for making this choice, although they are non-linearity specific. Such heuristics have helped to train networks with hundreds of layers [He et al., 2016]. Methods which scale weight initialisation by observing activation and gradient scales have also been effective [Mishkin and Matas, 2016]. Alternatively, if the network was previously trained, even for a completely different task, initialising with the pre-trained weights can be beneficial [Yosinski et al., 2014]. In the absence of pre-trained weights, an alternative is training a shallow network and then adding new layers until the desired depth is reached [Simonyan and Zisserman, 2015].

To train neural networks deeper than, for example, 20 layers requires structural changes beyond regularization, activation functions, and initialisation. The main goal of such structural changes is to reduce the shortest path through the network from input to output while keeping the maximal path the same length. This can produce networks that are both end-to-end trainable and able to take advantage of the richer function space offered by hundreds of layers. The two main techniques for achieving this highway connections [Srivastava et al., 2015] and residual connections [He et al., 2016] are similar. Highway connections are inspired by the gating mechanisms used in recurrent neural networks: for each layer a neural network with sigmoid activation learns to trade-off between skipping the layer or passing forward the layers transformation. At the start of training the network is usually biased towards skipping layers

to avoid vanishing gradients. Residual connections are simpler than highway connections as they don't involve adding new parameters to the network, instead a layer's input is added to its output before the non-linearity is applied. This introduces a shortcut that effectively skips the layer entirely. With residual connections the layers can be intuitively understood to be learning the residual of their original transformation. Here the term layer is used broadly, since residual connections often skip over several layers, adding the input of the first layer to the output of the last layer. Both highway connections and residual connections allow networks hundreds of layers deep to be trained. While highway connections are a more general solution residual connections have so far been applied more extensively [Hu et al., 2017a; Szegedy et al., 2017; Xie et al., 2017].

Learning large CNNs from scratch is computationally expensive, often taking weeks on multi-GPU systems [Simonyan and Zisserman, 2015; Szegedy et al., 2015]. Fortunately, the neural network architecture makes it easy to fine-tune a pre-trained model for a related task. CNNs have been shown to construct simple features in early layers, such as edge or corner detectors and more complex domain specific features, such as face detectors, in later layers [Zeiler and Fergus, 2014]. This advocates fine-tuning only the later layers, or even the last layer when the domains are similar, an approach that works well in practice [Razavian et al., 2014; Yosinski et al., 2014]. Networks trained on ImageNet have been fine-tuned for tasks such as scene recognition, fine grained animal classification, and pose estimation [Pfister et al., 2014; Donahue et al., 2014].

2.2 Language Generation and Recurrent Neural Networks

Natural language generation is key to automatic image captioning as it constitutes the final stage of the process: realising the selected visual concepts into a linguistic form. For generating styled image captions, the language generation stage is the focal point where linguistic style needs to be incorporated. The following gives a broad overview of language generation before focusing in on the recurrent neural network models frequently used in this thesis.

Approaches to language generation fall into two categories: template filling and generative language modelling. We briefly discuss template filling before covering generative language modelling – in particular neural network based generative language modelling – in more depth.

Template filling typically involves specifying the words that fill a pre-defined sentence structure; for example, the structure NP, VP, NP could be filled with “*the dog*”, “*ate*”, “*the grass.*”, where NP is a noun chunk, VP is a verb chunk, and NP is a noun

chunk. Far more complex templates are possible, but they are generally defined by human experts. To produce grammatically correct output, a text realiser [Gatt and Reiter, 2009] is typically applied to the filled templates. This uses rules and/or statistical techniques to choose the correct inflected word forms, enforce noun-verb agreement, and apply appropriate orthography. Because of the structured set of rules, generated text tends towards a similar structure, making it appear formulaic or robotic. However, this can be an advantage in domains such as weather forecast generation. Reiter et al. [2005] demonstrated annotators preferred computer generated weather forecasts to human generated variants, which they claim is in part due to the consistent generation rules designed to avoid ambiguity.

Generative language modelling is a collection of statistical techniques for learning linguistic relationships. Typically, we are interested in the probability of the next word in the sequence x_{i+1} given the previous words in the sequence x_1, \dots, x_i , contextual information c and parameters θ .

$$P(x_{i+1}|x_i, \dots, x_1, c; \theta) \quad (2.11)$$

n-gram language models [Shannon, 1951; Jurafsky and Martin, 2014] are one common choice for describing this probability distribution. Under the n-gram model we make a Markov approximation: $x_{i+1} \perp\!\!\!\perp x_{i-n}, \dots, x_1 | x_i, \dots, x_{i-n+1}$, with $\perp\!\!\!\perp$ denoting conditional independence. With this approximation and value for n (typically $1 < n < 6$), the most likely sequence can be computed efficiently with the viterbi [Viterbi, 1967] dynamic programming algorithm. If n and the vocabulary is large, then approximate search techniques such as beam search may be applied for efficiency reasons [Jurafsky and Martin, 2014]. Learning the conditional probability distributions $P(x_{i+1}|x_{i-n+1}, \dots, x_i)$ can be performed efficiently by counting occurrences of $x_{i-n+1}, \dots, x_i, x_{i+1}$ in the training data. To deal with data sparsity, it is often necessary to smooth these counts and apply back-off [Kneser and Ney, 1995]: where shorter n-grams are used if their longer counterparts were not seen in training. Context c can be incorporated by changing the form of the conditional probability distributions, although this is inefficient if c is continuous or high dimensional.

Recently, Recurrent Neural Networks (RNN) [Elman, 1990; Mikolov et al., 2010] have been used to build state-of-the-art language modelling systems [Zilly et al., 2016; Ha et al., 2017; Kim et al., 2016; Kuchaiev and Ginsburg, 2017]. RNNs do not rely on the Markov approximation. Instead, the previous words in the sequence are encoded as a continuous hidden vector. Because of the effectively infinite history, exact decoding is intractable and approximate search must be used. Beam search, and greedy decoding are common strategies. Context c can be incorporated by first embedding it into a vector space, then either: using it to initialise the hidden state of

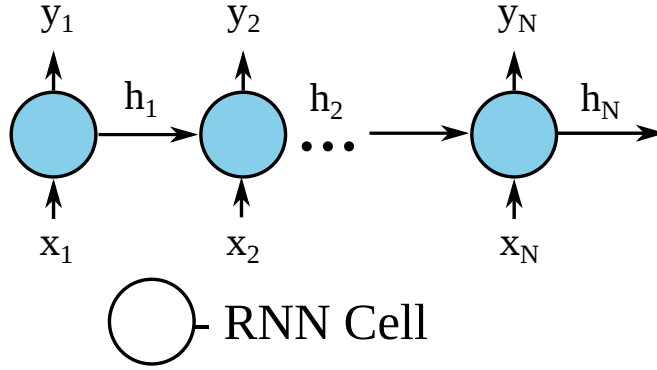


Figure 2.1: A simple Recurrent Neural Network. The recurrent layer R is duplicated, to match the length of the input $\{x_1, x_2 \dots x_N\}$ and output $\{y_1, y_2 \dots y_N\}$ sequences.

the RNN, pre-pending it to the RNN input sequence, or concatenating it with each input word embedding. A differentiable model used to extract c (such as a CNN in image caption generation) can easily be updated via back-propagation, allowing full joint training of context and language model.

2.2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [Elman, 1990] are a method of dealing with discrete-time input and output sequences in a neural network framework. They have been applied to domains including text generation [Wu et al., 2016; Fidler and Goldberg, 2017], video classification [Ng et al., 2015], speech recognition [Ng et al., 2015], and dynamic learning rate tuning for machine learning [Andrychowicz et al., 2017]. Their applications to text generation are most important to this thesis and are the focus of this review.

RNNs – Figure 2.1 – have an internal layer R , called the recurrent layer, that is unrolled via duplication to match the length L of the input sequence $X = \{x_1, x_2 \dots x_N\}$. Duplicate recurrent layers share parameters, ensuring a constant parameter count regardless of sequence length. This weight sharing eases the learning of sequential regularities; for example common bi-grams in text are not tied to a particular index in the sentence. Without weight sharing they would have to be learnt separately for all indexes. Variation across time is encoded with a hidden vector h_t passed between the recurrent layers. This acts as a memory of the previously seen sequence elements. The RNN outputs a sequence $Y = \{y_1, y_2 \dots y_N\}$, the same length as the input. For variable length sequences with mini-batch training, padding may be used.

Differences in the form of an RNN's recurrent layer lead to different RNN flavours, the most common of which are Elman [Elman, 1990], Long Short-Term Memory

(LSTM) [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Unit (GRU) [Cho et al., 2014a]. Elman networks were the first proposed and have the simplest form,

$$\begin{aligned} \mathbf{h}_t &= \sigma_h(W_x \mathbf{x}_t + W_p \mathbf{h}_{t-1}) \\ \mathbf{y}_t &= \sigma_y(W_h \mathbf{h}_t) \end{aligned} \quad (2.12)$$

Where weights W_x , W_p and W_h parameterise the recurrent layer. While it is common to explicitly add biases, they are omitted for compactness since they can be incorporated into the weight matrices by appending fixed unit dimensions to h_t and x_t . In practice, Elman networks suffer from vanishing and exploding gradients; recent work mitigates this, at the expense of expressivity, by enforcing orthogonal weight matrices [Vorontsov et al., 2017].

Another solution to the vanishing and exploding gradients problem is the Long Short-Term Memory cell, originally proposed by Hochreiter and Schmidhuber [1997] and further improved by other authors [Gers et al., 2000; Graves and Schmidhuber, 2005]. In its modern form [Graves, 2013], the LSTM separates the memory into hidden \mathbf{h}_t and cell \mathbf{c}_t vectors, and controls access to this memory with gates $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$. The basic LSTM cell can be written as (omitting bias vectors),

$$\begin{aligned} \begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \odot \left(\begin{bmatrix} \mathbf{W}_{xi} & \mathbf{W}_{hi} \\ \mathbf{W}_{xf} & \mathbf{W}_{hf} \\ \mathbf{W}_{xo} & \mathbf{W}_{ho} \\ \mathbf{W}_{xc} & \mathbf{W}_{hc} \end{bmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{pmatrix} \right) \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh \mathbf{c}_t. \end{aligned} \quad (2.13)$$

With weight matrices W and σ the sigmoid function. We give Equation 2.13 in block matrix form because it is the most efficient to implement on modern hardware. The LSTM has many different variants, such as peep-hole connections [Gers and Schmidhuber, 2000]. For an in-depth comparison of different variants see Greff et al. [2016].

Another RNN cell that has seen widespread use is the GRU [Cho et al., 2014a,b]: a simpler variant of the LSTM that lacks cell memory. It is an attractive model choice because it performs similarly to the LSTM in most problems [Jozefowicz et al., 2015], while being easier to implement and less computationally expensive [Cho et al., 2014b]. The standard GRU can be written as,

$$\begin{aligned} \mathbf{r}_t &= \sigma(x_t W_{x,r} + h_{t-1} W_{h,r} + b_r) \\ \mathbf{u}_t &= \sigma(x_t W_{x,u} + h_{t-1} W_{h,u} + b_u) \\ \tilde{\mathbf{h}}_t &= \sigma(x_t W_{x,c} + \mathbf{r}_t \odot (h_{t-1} W_{h,c}) + b_c) \\ \mathbf{h}_t &= (1 - \mathbf{u}_t) \odot h_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t \end{aligned} \quad (2.14)$$

With weights W , biases b , reset gate r , update gate u and σ the logistic function. We could also write Equation 2.14 in a block matrix form similar to Equation 2.13, although the dependence between \tilde{h}_t and r_t makes the representation less clear.

There are a few practical considerations for implementing RNN models. RNN can suffer from exploding gradients, so many authors clip gradients to a pre-defined range (typically $[-5, 5]$) – as suggested by Graves [2013]. When run in mini-batch mode, on variable sequence lengths, it is necessary to introduce padding and mask the loss function. Grouping similar length sequences together can avoid un-necessary calculations. Dropout and similar strong regularisation techniques should be applied to the input and output of the recurrent cell but not between recurrent layers as this can effect the memory capacity of the network [Zaremba et al., 2014]. Stacking RNN in several layers [Fernández et al., 2007; Graves et al., 2009] has proven to be an effective technique for improving RNN model performance by adding additional layers of abstraction from the input sequences. When greater model complexity is necessary it should be considered as an alternative to increasing the number of neurons in each cell.

Other forms for the recurrent layer have been proposed [Bayer et al., 2009; Zoph and Le, 2017], though they have not seen wide spread adoption, perhaps because the ubiquitous LSTM and GRU give good empirical performance over a range of different problems [Greff et al., 2016; Jozefowicz et al., 2015], and are therefore an easily justified model choice.

The standard way to train an RNN is to use Back-Propagation Through Time (BPTT) [Robinson and Fallside, 1987]. The recurrent cells are unrolled to the length of the sequence and then back-propagation is used to update all copies of the recurrent cell. Rather than defining a new set of weights for each copy, a link to the original weights allows efficient aggregation of updates. Standard BPTT is computationally expensive when the sequence is long; truncated BPTT [Williams and Peng, 1990] presents a practical alternative. In this method, back-propagation is run every k_1 time steps on the last k_2 time-steps – where k_1 and k_2 are hyper-parameters. Truncated BPTT enables training on very long sequences without needing to split them.

RNNs have difficulty generating long sequences because generation errors compound, and they forget over time. One possible solution is the hierarchical RNN [Lin et al., 2015; Chung et al., 2017]: multiple-layers of RNNs running at different time scales. The long time-scale units give long term stability and structure, while the short time-scale units focus on the details such as the next word or character. These types of models have a long history [Schmidhuber, 1992; El Hihi and Bengio, 1996; Lin et al., 1996], but have only recently come to prominence as neural network language models become state-of-the-art. For their Clockwork RNN, Koutnik et al.

[2014] develop a new RNN cell where the neurons are partitioned into layers that change at different pre-defined time scales. Lin et al. [2015] use an RNN with a single cell for each sentence, and an RNN with a cell for each word. The word-level RNN is trained first before being fixed and used to train the sentence-level RNN. Chung et al. [2017] implement an RNN stack that learns how to break sequences into hierarchical chunks, to avoid specifying the time scales of the RNNs. At each time step, every layer selects one of three operations: *UPDATE* which is the standard RNN operation, *COPY* which copies the unmodified hidden state from the last time step, and *FLUSH* which passes the current state to the layer above and resets the current state. Using the operations, the model appears to learn boundaries at the word, syntactic, and semantic levels.

2.2.2 Sequence-to-Sequence Models

Sequence-to-sequence models form a core component of Chapters 5 & 6. In Chapter 5, a sequence-to-sequence model is used as a means of sentence re-writing, while in Chapter 6 a sequence-to-sequence model generates image captions from an ordered sequence of terms. In both cases RNN sequence-to-sequence models are employed. This section provides background on this type of model.

When a machine learning model takes one or more sequentially structured inputs, and generates a sequentially structured output, it can be called a sequence-to-sequence model. This is a broad class of model, but we restrict our discussion to sequence-to-sequence models utilising RNNs [Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014; Wu et al., 2016]. In the language case, which we are primarily concerned with, both sequences are typically discrete tokens such as words or parts-of-speech tags. Sequence-to-sequence models have been applied to machine translation [Kalchbrenner and Blunsom, 2013], sentence compression [Rush et al., 2015; Auli and Rush, 2016], captioning videos [Venugopalan et al., 2015] and dependency parsing [Zhang et al., 2017b].

RNN based sequence-to-sequence models consist of an encoder RNN for embedding the input sequence, and an RNN decoder for generating a new sequence from this embedding. One popular option for embedding [Sutskever et al., 2014] is to use the last hidden output from the encoder $h_{enc,M}$ to initialise the decoder hidden state $h_{dec,0} = h_{enc,M}$, where $h_{dec,0}$ is the initial decoder hidden state. Alternatively, $h_{enc,M}$ can be concatenated with input word embeddings, or attention can be used – see Section 2.2.2.1 for details.

Training a sequence-to-sequence model requires paired input and output sequences. Typically, the loss function is applied only to the outputs of the decoder,

and the errors propagated back to the encoder through the copied hidden state. This can lead to difficulty in training [Bahdanau et al., 2014] as the effective depth of the network can be as long as the input sequence. In language tasks, reversing the input sequence has proved beneficial [Sutskever et al., 2014] because on average it reduces the effective depth: the last token input to the encoder will often align to the first token output by the decoder. However, by far the most common approach is to use a bidirectional encoder [Schuster and Paliwal, 1997a; Sundermeyer et al., 2014], which can be seen as two encoders one running forwards over the input sequence and another running backwards. This allows language to be process in the correct order while still reducing the effective depth of the network.

2.2.2.1 Attention

Early neural network sequence-to-sequence models performed poorly on long sequences. Training is made difficult by a large effective network depth and requiring that the entire sequence be embedded in a fixed size vector. Attention [Bahdanau et al., 2014] was proposed as a solution, and has since become an integral part of sequence-to-sequence modelling. Intuitively, attention is a soft word-level alignment between the input and output sentences. By directly connecting the hidden outputs of the encoder cells to the decoder, attention effectively reduces network depth. It also avoids compressing the entire input sequence into a fixed length vector; instead, the embedding grows to match the input sequence length [Bahdanau et al., 2014].

Several different forms of attention have been proposed for sequence-to-sequence models [Bahdanau et al., 2014; Luong et al., 2015]; however, their many similarities permit a concise summary. Given hidden outputs from the encoder $h_{enc,i}$, $i \in 1 \dots L_{enc}$ and decoder $h_{dec,j}$, $j \in 1 \dots L_{dec}$, we can define a similarity function $g(\cdot)$ with trainable parameters θ :

$$a_{i,j} = \frac{1}{k} g(h_{enc,i}, h_{dec,j}; \theta) \quad (2.15)$$

Where k is a normalisation constant, ensuring $\sum_{i=1}^{L_{enc}} a_{i,j} = 1$. The variable $a_{i,j}$ is referred to as the attention and is used to weight the encoder hidden states when calculating the context vector c_j .

$$c_j = \sum_{i=1}^{L_{enc}} a_{i,j} h_{enc,i} \quad (2.16)$$

This context vector is then used to generate the next output term by combining it with $h_{dec,j}$, and either feeding it into the output layer $h_{out,j} = c_j \oplus h_{dec,j}$ [Luong et al., 2015] or into the next cell of the RNN $\tilde{h}_{enc,j} = c_j \oplus h_{enc,j}$ [Bahdanau et al., 2014]. Here \oplus is either concatenation or element-wise addition.

If the context vector is provided directly as input to the output layer, then the attention has no effect on the hidden state of the RNN – except indirectly through discrete output samples – and so no memory of previous alignments is retained. This negatively impacts accuracy on some problems [Luong et al., 2015], but may be more computationally efficient because the RNN and attention components can be represented compactly as batched matrix multiplications.

There are three main variants for similarity function $g(\cdot)$. The simplest is the dot product $g(h_{enc,i}, h_{dec,j}) = h_{enc,i} \cdot h_{dec,j}$. An alternative is a bi-linear map, which is effectively a weighted dot product $g(h_{enc,i}, h_{dec,j}) = h_{enc,i} W_a h_{dec,j}$ with learnable weight matrix W_a . The most complex of the three is a non-linear mapping, such as a multi-layer perceptron [Wu et al., 2016], applied to the concatenated embeddings $g(h_{enc,i}, h_{dec,j}) = \sigma(W_a[h_{enc,i}; h_{dec,j}])$, with non-linearity $\sigma(\cdot)$.

Many extensions have been proposed to improve attention models. Local attention models attempt to up-weight more likely matchings [Luong et al., 2015]. For example, in most language translation problems the i 'th input word is most likely to match the j 'th input word. We can impose soft or hard constraints by weighting and re-normalising $a_{i,j}$ with the equation $a'_{i,j} = \frac{1}{k'} \alpha(i, j) a_{i,j}$ with weighting function $\alpha(i, j)$ and normalisation constant k' . Local attention is not widely used because it reduces the flexibility of the attention layer, often leading to weaker performance.

In some cases, RNN language model decoders will repeat themselves, this can be mitigated with self attention [Paulus et al., 2018; Xia et al., 2017; Shao et al., 2017], where attention is applied over previous decoder hidden states $h_{dec,1}, \dots, h_{dec,j-1}$ in addition to the encoder hidden states. Similarly, attention may repeat, so a soft constraint encouraging attention over all hidden states is sometimes used to ensure coverage of the input sentence [Paulus et al., 2018].

Attention can be used to copy words directly from the input sentence to the output sentence [Luong et al., 2014], if we interpret it as a soft form of word-word alignment. In a translation context, this can allow previously un-seen words, such as proper nouns, to be copied un-translated. However, attention as an alignment approach has been generalised and is more broadly applicable [Vinyals et al., 2015a]. In Chapter 5 I adapt this copying approach for sentence simplification.

2.3 Image-Caption Generation

This section reviews descriptive image-caption generation approaches. These approaches form a basis for the styled caption generation techniques developed in this thesis. The neural network approaches in Section 2.3.3 are the most relevant. Other approaches are described to give context. Section 2.3.4 defines several descriptive

caption evaluation metrics that are used throughout this thesis. Caption descriptiveness metrics are used to ensure styled captions still relate to the image.

Automatically producing image captions that are both natural and relevant is a difficult problem. Two essential components: visual concept detection, reviewed in Section 2.1, and Natural Language Generation (NLG), reviewed in Section 2.1, are both research problems in their own right.

Approaches to caption generation fall into three rough, and non-mutually exclusive, categories: caption retrieval [Farhadi et al., 2010; Ordonez et al., 2011; Hodosh et al., 2013; Karpathy et al., 2014a], object detection and generation [Li et al., 2011; Kulkarni et al., 2011; Yang et al., 2011], and deep generative networks [Donahue et al., 2015; Karpathy et al., 2014a; Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015b]. Caption retrieval approaches combine captions or caption segments matching the query image. Detection and generation approaches use a multi-stage pipeline involving object detection, content planning and language generation. Deep generative networks are typically end-to-end trainable neural networks using both convolutional and recurrent components. Current state-of-the-art approaches follow this framework, though caption retrieval is surprisingly competitive [Devlin et al., 2015].

For a comprehensive review of automatic image caption generation before 2016 see Bernardi et al. [2016].

2.3.1 Caption Retrieval

Caption retrieval is an image captioning technique that relies on the idea that similar images will have similar captions. The general approach is to find captions relevant to the target and compose them to form a new caption. Relevant captions are retrieved by either searching for images similar to the target, and collecting their paired captions, or defining an image-caption similarity metric. Caption retrieval is non-parametric, potentially allowing high output diversity. Generated captions may even sound natural, since retrieved captions were wholly or in part written by human annotators. However, retrieval is also limited by the captions available in the retrieval set: to get precise captions composition is a necessity.

Some approaches retrieve complete captions and transfer them to the target image [Farhadi et al., 2010; Ordonez et al., 2011; Mason and Charniak, 2014; Yagcioglu et al., 2015; Devlin et al., 2015]. Farhadi et al. [2010] map images and captions to a discrete intermediate space composed of (object, action, scene) tuples. Images are matched to training sentences that have similar sets of predicted tuples. Ordonez et al. [2011] match target image to similar training images using low level visual fea-

tures, object detections and scene detections. The captions of each of the matched images are added to a pool, then the most relevant caption is selected by matching words with object detector scores. Even with 1 million images the captions are often imprecise because they were transferred directly from another image. Yagcioglu et al. [2015] find similar images in the training set using CNN features. They then calculate a weighted average of word vectors from the associated captions. The closest candidate caption to the average is transferred to the image. Mason and Charniak [2014] find similar images in the training set and then model the probability of a word given the query image $p(w|I)$ using Bayes rule $p(w|I) \propto p(I|w)p(w)$. They estimate $p(I|w)$ using density estimation on the k-nearest-neighbours, and $p(w)$ with global uni-gram frequency. Using $p(w|I)$, the captions are ranked and the highest scoring one is transferred.

Another approach to caption transfer is to learn a joint image-caption vector space in which image embeddings are close to their caption embeddings. Given a new image, cross-modal retrieval is performed by embedding the image and finding the closest captions. Hodosh et al. [2013] use Kernel Canonical Correlation Analysis (KCCA) to find a maximally correlated linear projection of images and text into a common space. Being a kernel method only similarity functions need defining, while the joint space is implicit. Unfortunately for large datasets, KCCA suffers from prohibitively high memory usage. Socher et al. [2014] embed sentences using a recursive neural network defined over automatically generated parse trees (called DT-RNN); images are embedded as CNN features. They minimise the following pair-wise ranking loss:

$$\begin{aligned} \mathcal{L}_{\mathcal{I}, \mathcal{J}, \mathcal{I}', \mathcal{J}'} = & \max(0, \Delta - (W_I v_I)^T v_s(\theta) + (W_{I'} v_{I'})^T v_s(\theta)) \\ & + \max(0, \Delta - (W_I v_I)^T v_s(\theta) + (W_{I'} v_{I'})^T v_{s'}(\theta)) \end{aligned} \quad (2.17)$$

Where I , and s are a correct image caption pair and s' and I' are noise contrastive samples. The margin is Δ , the parameters of the DT-RNN are θ with output embedding $v_s(\theta)$, W_I is an image embedding projection matrix, and v_I is the image embedding. Karpathy et al. [2014a] expand this idea to embedding sentence fragments and image patches into a joint space. Alignment between fragments and patches is achieved with multiple instance learning [Andrews et al., 2003], assuming each sentence fragment has at least one associated image patch. Their sentence fragments are dependency parse triples embedded using a simple non-linear function, while their image features are extracted with a Region Convolutional Neural Network (RCNN) [Girshick et al., 2014].

Rather than transferring entire captions to the query image, some authors [Gupta and Mannem, 2012; Kuznetsova et al., 2012, 2014; Lebet et al., 2015] have composed

captions from multiple candidates, with the aim of increasing caption specificity. Gupta and Mannem [2012] combine multiple captions by distilling them into an intermediate representation of ((determiner, attribute, subject), verb, preposition, (determiner, attribute, object)). The elements of this representation are chosen through frequency counting token n-grams. The final text is realised using SimpleNLG [Gatt and Reiter, 2009], which handles aspects such as syntax and morphology. Kuznetsova et al. [2012] glue together relevant phrases using integer linear programming, and in a later work, parse trees for handling long distance relations [Kuznetsova et al., 2014]. They also generalise captions as a pre-processing step by removing irrelevant details or sub-strings that lack visual grounding. The results demonstrate improvement over full caption retrieval with respect to human judgements. Lebrete et al. [2015] map CNN image features and phrases into a joint space. At test time they retrieve phrases and compose them with a tri-gram language model.

2.3.2 Object Detectors and Text Generators

Detection and generation methods use a pipeline of individual components, the general structure of which is similar across systems. Object and attribute detectors identify visual concepts, which are then mapped into an intermediate representation where smoothing is applied to enhance coherency. A natural language generation system realises the smoothed intermediate representation as text. The methods presented here bare a strong resemblance to the neural network models in Section 2.3.3; however, unlike the neural network models they cannot be trained end-to-end.

The concept detectors used in these approaches are typically the state-of-the-art at the time of publication. For scene and attribute concepts it is common to train new classifiers on low level features [Li et al., 2011; Mitchell et al., 2012]. For objects, existing classifiers [Li et al., 2011; Kulkarni et al., 2011; Yang et al., 2011; Mitchell et al., 2012] such as deformable parts models [Felzenszwalb et al., 2010] are common. Often a fixed set of classes is chosen based on the dataset being used, e.g the UIUC PASCAL sentence dataset [Rashtchian et al., 2010] has 20 object classes. More recent approaches use CNNs [Fang et al., 2015]. The choice of classes is often strongly related to the intermediate semantic representation – typically one classifier or group of classifiers is needed for each concept type (eg objects, actions, attributes or scenes).

The intermediate semantic representations are a key difference between models. Yang et al. [2011] represent an image as two nouns, a verb, a preposition and a scene, while Kulkarni et al. [2011] build a graph of adjective-noun pairs linked by prepositions. Fang et al. [2015] use an unordered list of words with visual detection probabilities.

To filter out classifier errors and ensure consistent semantics, smoothing is typically applied to the intermediate semantic representation; graphical models are a popular choice. Kulkarni et al. [2011] and Rohrbach et al. [2013] employ Conditional Random Fields (CRFs) with potentials learnt from external text corpora; Yang et al. [2011] use Hidden Markov Models (HMMs) trained on text and Farhadi et al. [2010] use Markov Random Fields. Alternatively, by choosing subject-object-verb triples as the intermediate representation, Krishnamoorthy et al. [2013] can use an n-gram language model for ranking. With a specialised model it is also possible to generate language without smoothing the intermediate representation [Fang et al., 2015].

Approaches to NLG are typically based on template filling or n-gram language models [Yang et al., 2011; Kulkarni et al., 2011; Krishnamoorthy et al., 2013], though off-the-shelf machine translation systems such as Moses [Koehn et al., 2007] have also been used [Rohrbach et al., 2013]. Krishnamoorthy et al. [2013] use a template that they fill with semantic tuples and additional words chosen by a language model. Yang et al. [2011] use a set of rules to define a flexible template that they fill using semantic tuples and a language model. Their results show this model matches ground-truth captions more accurately than less flexible template filling. Kulkarni et al. [2011] compare simple template filling with ordered semantic words connected via function words generated from an n-gram language model. Human evaluators preferred the template filling variant even though it was a weaker match to the ground-truth captions than the language model variant. Evaluations of language model vs template filling approaches by Li et al. [2011] came to similar conclusions, though the language model sentences were judged as more creative. Mitchell et al. [2012] improve upon template filling by using a complex set of rules to generate, populate, and merge syntactic trees. Fang et al. [2015] achieve results competitive with the best end-to-end neural network systems using high accuracy CNNs for selecting descriptive words, a custom maximum entropy language model with beam search decoding and re-ranking, rigid word restrictions, and a large image-caption dataset.

The detection and generation methods reviewed here are not yet at the level where they compete with human annotators. One issue is the constrained concept domain, another is the concept detectors with fixed language realisations. CNN detectors [Krizhevsky et al., 2012] have since helped to alleviate the constrained concept domain, because they can reliably identify thousands of concepts. The other issue, constrained vocabulary for visual objects, is partially addressed in this thesis (Chapter 3) and associated publications [Mathews et al., 2015]. Even still generated captions can be formulaic and un-natural when they rely on rigid intermediate representations and template surface realisation. This problem is partially addressed by

the end-to-end trainable neural network models in the following section.

2.3.3 End-to-End Neural Network Models

Deep generative networks are similar to the detection and generation methods of the previous section. We choose to make the distinction based on the degree of end-to-end learning. The deep generative networks replace some or all of the detection and generation components with end-to-end systems. Enabling this are more flexible models, and new datasets including hundreds of thousands of captions and images [Hodosh et al., 2013; Young et al., 2014; Chen et al., 2015]. Previously datasets consisted of only a few thousand images [Rashtchian et al., 2010], which is too small for end-to-end learning to be effective, or were mined from web resources [Ordonez et al., 2011], where latent context plays a major role in image descriptions.

Deep generative image captioning models typically consist of a CNN (Section 2.1.2) for object detection combined with an RNN (Section 2.2.1) for text generation [Donahue et al., 2015; Karpathy et al., 2014a; Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015b]. These two components can be composed and learnt jointly through back-propagation. The second last layer of the CNN – which can be seen as a set of high level visual features – is connected to the RNN, either through the RNN’s input path [Vinyals et al., 2015b; Donahue et al., 2015], hidden state [Jin and Nakayama, 2016], or via a custom addition to the RNN cell. Mao et al. [2015] join the RNN and CNN at the output of the RNN, which they claim reduces the demands on the recurrent layer, allowing it to focus on language generation. Although initially promising this design has not caught on, perhaps because larger datasets and more computational power allows for high capacity recurrent layers. Donahue et al. [2015] generalise the CNN+RNN model to: sequential input, single output (action recognition in video); single input, sequential output (image captioning); and sequential input, sequential output (video descriptions). Their use of the LSTM cell has become the de-facto standard. Vinyals et al. [2015b] develop an CNN+RNN model for static captions and present a thoughtful analysis of different CNN+RNN training and decoding techniques, such as curriculum learning and beam search. Their system has had a large impact on the literature because of state-of-the-art results at publication, publicly released code, several updates [Vinyals et al., 2017], and media exposure.

One possible problem with CNN+RNN models is the compression of all image semantics into a fixed length vector – typically the second last layer of a CNN. Xu et al. [2015a] demonstrated that localised features from earlier layers of the CNN could be exploited using an RNN with attention. Their soft attention model is fully differentiable, similar to attention in neural machine translation (Section 2.2.2.1); their

hard attention model can be seen as a reinforcement learning approach. An extra regularization term encourages attention to all parts of the image. This approach improved BLEU and METEOR over CNN+RNN models, and learnt a correspondence between parts of the caption and spatial locations in the image. Chen et al. [2017] extend the spatial attention model by enabling channel wise attention. This allows the model to fixate on important semantic features.

CNN+RNN models are typically trained to minimise the negative log-likelihood of the caption given the image; evaluation then uses a metric such as BLEU, METEOR or CIDEr. To achieve the best performance, it helps to use the final evaluation metric to inform learning [Ranzato et al., 2015; Rennie et al., 2016; Zhang et al., 2017a], though this does not always translate to more favourable human evaluations [Wu et al., 2016]. Many popular metrics are non-differentiable, so cannot be used in the loss function directly; however, reinforcement learning [Sutton and Barto, 1998] is a potential solution. Word choices are defined as actions, and the reward is a function of the chosen metric – evaluated once the entire sequence is generated. The space of word choices is extremely large, making learning difficult. In response, Ranzato et al. [2015] pre-train with log-likelihood and then with a mixed REINFORCE [Willia, 1992] and log-likelihood objective. Building on this idea, Rennie et al. [2016] introduce self-critical sequence training using the argmax decoding as a reward baseline, reducing the variance of the expected gradient and leading to state-of-the-art image captioning results. Actor-critic reinforcement learning has also proven effective [Zhang et al., 2017a].

One line of inquiry that has recently improved image caption generation is the use of high level concepts [Fang et al., 2015; Wu et al., 2015; Gan et al., 2017b]. In a standard CNN+RNN model, the last fully connected layer is removed and the features from the CNN are passed directly to the RNN – a process that discards high level information encoded in the last layer. Retaining the high level information reduces the required capacity of the RNN and adds a short training path between text concepts and image features [Liu et al., 2016]. Wu et al. [2015] use a region based multi-label CNN fine tuned on common MSCOCO terms to define image semantics. Proposals from different regions are max-pooled and provided to the language, generating LSTM on the first time-step. Fang et al. [2015] use a region based multi-label CNN, fine tuned on MSCOCO via multiple instance learning [Zhang et al., 2005]. With a maximum entropy language model and sentence re-ranking they are able to achieve comparable performance to Wu et al. [2015]. It is interesting to note that even though LSTM models are the current trend, the more traditional maximum entropy language model can perform similarly; however, tuning may be more difficult for these model types. Gan et al. [2017b] introduce an LSTM with concept depen-

dent weight matrices generated by weighting a weight matrix for each concept by the probability that concept is detected. A factorised tensor representation is used to reduce the number of parameters and achieve state-of-the-art performance.

2.3.4 Evaluating Generated Captions

For the automatic evaluation of generated image captions, the most common metrics are BLEU [Papineni et al., 2002], Rouge-L [Lin, 2004], METEOR [Denkowski and Lavie, 2014], CIDEr [Vedantam et al., 2015], and SPICE [Anderson et al., 2016]. These metrics measure the similarity between generated captions and the ground truth, and so require at least one ground truth caption for each test image – the popular MSCOCO dataset [Chen et al., 2015] provides 5 ground-truth captions per test image. BLEU, ROUGE and METEOR were originally designed for use in machine translation and summarization tasks, but were adopted for image caption evaluation. CIDEr and SPICE are both specially designed for image caption evaluation, and so tend to correlate more strongly with human judgements of caption relevance.

BLEU [Papineni et al., 2002] is an n -gram precision metric, often denoted BLEU-1, BLEU-2, BLEU-3, or BLEU-4 to indicate the maximum n -gram length. The calculation of BLEU has three distinct stages: calculation of the modified n -gram precisions, averaging log n -gram precisions across different n , and applying the brevity penalty. Modified precision is the fraction of n -grams in the generated sentence that match n -grams in one of the test sentences, although duplicate n -grams in the generated sentence may only match up to the maximum count in any test sentence. For example, if “the” occurs three times in the generated sentence but at most twice in any test sentence, then only two occurrences of “the” match. In practice, the modified n -gram precision – denoted p_n – is calculated using all generated sentences as,

$$p_n = \frac{\sum_{C \in \{Generated\}} \sum_{gram_n \in C} Count_{clip}(gram_n, C)}{\sum_{C' \in \{Generated\}} \sum_{gram'_n \in C'} Count(gram'_n, C)} \quad (2.18)$$

Where: $Count(gram_n, C)$ is the number of occurrences of each n -gram – denoted $gram_n$ – in sentence C , and $Count_{clip}(gram_n, C)$ is the number of occurrences of each n -gram in the generated sentence with clipping so that it does not exceed the highest count in any ground truth sentence.

$$Count_{clip}(gram_n, C) = \min(Count(gram_n, C), \max_{G \in \{GroundTruth\}_C} Count(gram_n, G)) \quad (2.19)$$

BLEU can then be expressed as:

$$BLEU = b \exp\left(\sum_{i=1}^n \frac{1}{n} \log p_i\right) \quad (2.20)$$

Where b is the brevity penalty defined in Equation 2.21 by c the word count for all generated sentences and r the effective reference length: calculated by summing the number of words in the best matching (in terms of sentence length) ground truth sentence for each generated sentence.

$$b = \begin{cases} 1, & \text{if } c > r \\ \frac{e}{1-r/c}, & \text{if } c \leq r \end{cases} \quad (2.21)$$

ROUGE [Lin, 2004] comes in several different variants; the two most common are ROUGE-N and ROUGE-L. ROUGE-N is an n-gram recall based metric computed between the generated sentence and the set of ground-truth sentences. ROUGE-N is not particularly common for image caption evaluation. ROUGE-L is an f-measure, based on the longest common subsequence (of words) between the generated and ground truth captions. We present the variant of ROUGE-L that is used in the MSCOCO evaluation server as it is the most relevant to image captioning [Chen et al., 2015]. First, the precision and recall are calculated for each generated caption and the corresponding set of ground truth captions as:

$$R_{lcs} = \max_{G \in \{GroundTruth\}_C} \frac{1}{m} LCS(C, G) \quad (2.22)$$

$$P_{lcs} = \max_{G \in \{GroundTruth\}_C} \frac{1}{u} LCS(C, G) \quad (2.23)$$

Where C is the generated sentence of length u , G is a ground truth sentence of length m , and $LCS(C, G)$ is a function computing the length of the longest common subsequence between C and G . The sentence pair ROUGE-L score is then:

$$ROUGE-L = \frac{(1 + \beta^2)P_{lcs}R_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (2.24)$$

β is the standard parameter in the f-measure where the common F_1 score has $\beta = 1$. Values of β larger than 1 weight recall higher than precision, while the reverse is true for values of β less than 1. The MSCOCO evaluation server has $\beta = 1.2$. The final ROUGE-L score is the mean over all generated captions.

METEOR [Denkowski and Lavie, 2014] aligns generated captions with each of the ground truth captions at a word or phrase level. Matching words or phrases uses: exact string matching, word stem matching, synonym matching in WordNet, or paraphrase matching using an external paraphrase table. The alignment between sentences is constructed by a beam search that tries to: match each word at most once, maximise the number of matched words, minimise the number of separate contiguous matches. Precision and recall are calculated in a weighted fashion, with

separate weights for content vs function words and for each of the different types of phrase matcher (eg exact string matching or synonym matching). The precision P and recall R are combined in a harmonic mean F parameterised by α :

$$F = \frac{PR}{\alpha P + (1 - \alpha)R} \quad (2.25)$$

Finally the METEOR score is calculated using a chunk penalty Pen , defined in terms of the number of contiguous matching chunks c and the total number of matches m :

$$\text{METEOR} = (1 - \text{Pen})F \quad (2.26)$$

$$\text{Pen} = \gamma \left(\frac{c}{m} \right)^\beta \quad (2.27)$$

The parameters α, γ, β and word match weights were chosen using grid search to best correlate with human judgements on 8 separate language translation tasks.

CIDEr [Vedantam et al., 2015] is based on cosine similarity of TF-IDF vectors of n -grams computed on the generated sentence C and ground truth sentence G . TF-IDF [Sparck Jones, 1972] weights $g^n()$ are computed for n -grams up to length 4, where images – specifically the set of ground-truth captions relating to an image – are taken as ‘documents’ in the TF-IDF weighting. The CIDEr score for generated sentence C using n -grams of length n is denoted $\text{CIDEr}_n(C)$ and calculated as:

$$\text{CIDEr}_n(C) = \frac{1}{m} \sum_{G \in \{\text{GroundTruth}\}_C} \frac{g^n(C) \cdot g^n(G)}{\|g^n(C)\| \|g^n(G)\|} \quad (2.28)$$

The final CIDEr score is the average CIDEr_n score for $1 \leq n \leq 4$.

SPICE [Anderson et al., 2016] is a metric designed for evaluating the semantic relevance of generated image captions. It does not rely on n -gram overlap as many other metrics do; instead, generated and ground truth captions are both mapped to a graph-based semantic representation called a *scene graph*. This representation preserves objects, attributes and relations while stripping away lexical and syntactic details. A dependency parser is applied to establish relationships between the words in the caption, which are then mapped into object, relation, and attribute components. Following this, a set of rules is used to build the *scene graph*. To calculate the SPICE score, both the generated caption *scene graph* and the joint *scene graph* for all corresponding ground truth captions are decomposed into a set of tuples containing objects, attributes, and relations – these tuples are of length one, two or three. The precision and recall are calculated with these tuples, where matching tuple elements have either the same lemmatized word form or belong to the same synset. SPICE is the F_1 score computed with the precision and recall.

2.4 Defining Style

To generate image captions in a particular style, it first helps to have a solid definition of what constitutes style. This section outlines different definitions of style in order to clarify what style is, and to suggest how it may be concisely represented for modelling purposes.

In common usage, the word “*style*” either refers to a linguistic property that all written texts have, or to an aesthetic judgement (eg to say a text was dull and uninteresting one might say “the piece lacks style”). In this thesis we are concerned with the former usage, where style is a linguistic property of all written texts.

Incorporating style into automatic systems requires a relatively concise definition; ideally, providing clues on how to separate it out. One definition of style used in automatic style analysis is: how something is communicated, rather than what is communicated [Pennebaker and King, 1999; Tausczik and Pennebaker, 2010]. This implies a clear distinction between style and content. A complementary definition for style is: a set of consistent and distinguishable linguistic choices [Karlgren, 2005; Khosmood and Levinson, 2008; Verdonk, 2002]. Together, these definitions suggest style can be separated by controlling for content (what is communicated) and modelling consistent description variability (how it is communicated).

The literary discipline of stylistics takes a more nuanced view of style, recognising it as an integrated property affecting every level of language, from its shape on the page to the contextual meaning [Simpson, 2004]. Literary stylistics is mostly concerned with interpreting strategies writers use to shape a text’s meaning within its context [Simpson, 2004; Verdonk, 2002]. We can therefore extend the definition in the previous paragraph to: Style is a set of consistent and distinguishable linguistic choices that shape meaning within a context. This definition now encompasses cases where style can be said to affect what is communicated. Simpson [2004] uses the example “*That puppy’s knocking over those potplants!*” in which the speaker has chosen to use the word “*puppy*” to describe a young canine, although, words such as “*dog*” or “*animal*” would also preserve the truth of the sentence. In Simpson’s view, the choice of the word “*puppy*” could have been made because of the word’s positive connotations, rather than to change the semantic content. If this is the case and the word was chosen to shape the broader contextual meaning then it is a stylistic choice. To apply the idea of style as choice, in its entirety, requires high level understanding founded on contextual reasoning and common-sense knowledge beyond current computational techniques. However, realising that style affects semantics and connects with a broader context are useful insights when designing models and analysing stylistic variation.

While the above deals with a conceptual definition of style as a whole, a particular instance of style can be defined with a fixed set of attributes [Pavlick and Nenkova, 2015; Ficler and Goldberg, 2017] or implicitly with a document collection. There are many possible stylistic attributes, examples of which include: formality, complexity, rhythm, sentiment, voice and point-of-view. Implicitly defined style uses document collections such as those from a single author [Stamatatos, 2009], genre [Kiros, 2015] or text-type [Khosmood and Levinson, 2008]. Xu [2017] provides a useful summary of different types of styles, both implicit and explicit, that are of interest for automatic generation.

It is worth noting that genre and style are not synonyms nor are they entirely independent of one another. Genre is a type of communication with socially agreed upon conventions [Bawarshi and Mary Jo Reiff, 2010], that typically apply to both content and style. For example, the science fiction genre has content conventions encouraging authors to deal with the imagined concepts of advanced technologies, extra-terrestrials, time travel, and similar concepts. Whereas, the style conventions of science fiction, as a narrative genre, include: using technical language, forming a linear narrative, and using dialogue. Some of these style conventions such as forming a linear narrative and using dialogue are shared across many genre, including the literary genres romance and crime. Even though a subset of a genre's conventions may encourage or enforce particular style choices, there are often many style choices left up to the author. This leads to authors having distinctive styles within a genre.

2.4.1 Sentiment

Sentiment can be roughly defined as an expressed affective value, opinion, or emotion [Pang, 2006; Hovy, 2015]. This places sentiment firmly in the scope of subjective judgements that cannot be objectively observed or verified [Pang, 2006]. Typically sentiment is not considered an aspect of style as it is only an opinion or emotion. However, the way sentiment is expressed and whether to express it at all can be considered stylistic choices. In contrast many recent works in computer science consider sentiment polarities (ie positive, negative, and neutral) to be particular styles [Shen et al., 2017; Fu et al., 2018; Prabhumoye et al., 2018]. Thus we briefly review aspects of sentiment and sentiment analysis.

Sentiment is often modelled with a positive, negative or neutral valence and a numerical indication of strength. Where positive sentiment shows support or positive feelings towards a topic and negative sentiment shows disagreement or negative feelings towards a topic. While this model of sentiment appears relatively crude, it is frequently used for understanding public opinion towards products, movies, and

political entities.

When defining sentiment it is important to consider aspects to texts that are not considered sentiment. Often we do not consider objective statements such as “*the stock price rose*” to convey sentiment even if the objective statement is overwhelmingly considered positive, such as in the case of stock price rises [Pang and Lee, 2008]. Though, this is not unilaterally accepted, for example classifying news articles as “good news” or “bad news” [Koppel and Shtrimberg, 2006] has been considered a sentiment classification task. Moreover, without fine grained supervision it can be difficult to distinguish between objective statements with strong connotations and subjective statements expressing an opinion [Pang and Lee, 2004; Balahur and Steinberger, 2009]. Thus many practical systems do inadvertently include such objective statements when analysing sentiment. There are a number of opinion related text classification tasks that are not, in general, considered sentiment classification tasks, such as classifying documents by political leaning (eg liberal vs republican), or classifying predictive opinions (eg candidate *A* is likely to win the election) [Pang and Lee, 2008].

A substantial amount of work has been put into sentiment analysis in the context of reviews [Pang and Lee, 2008], where opinions are communicated in text as well as via ordinal (eg star rating) or binary (eg like vs dislike) attributes. This provides a ready-made labelled data-set with the ordinal or binary attribute forming the label. While machine learning systems trained on such data can work relatively well, they loose fine-grained sentiment, or varying sentiment towards different aspects of the topic eg “*I think children would enjoy this film but I hated it...*”. There is also the issue of different reviewers having different rating scales [Hovy, 2015], for example one reviewer may never give less than 3 stars while another may frequently give 1 star reviews.

Hovy [2015] argues that current sentiment analysis research does not sufficiently consider the nuances of sentiment. First sentiment can be classified based on whether it corresponds to opinions, or feelings and emotions. Next, the positive, negative, and neutral categories are too general, rather the author suggests using positive, negative, mixed (both positive and negative sentiment expressed e.g. “*Sometimes I like milk in my coffee, other-times I hate it.*”), neutral (neutrality or lack of opinion is stated e.g. “*I don’t mind if my coffee has milk or not.*”), or unstated (sentiment is alluded to but not stated e.g. “*I have strong feelings about milk in coffee.*”).

2.4.2 Style Usage and Effect

Linguistic style is an essential part of written communication; by generating captions with a strong linguistic style we aim to reach a broader audience, reduce misinformation, and engaging viewers. This section reviews research from psychology and linguistics to identify the range of effects style has, and in doing so reveals the potential applications for style in image captioning.

Linguistic style has been shown to reflect the personality of the author [Pennebaker and King, 1999; Oberlander and Gill, 2006] and to affect the behaviour of the readers. Oberlander and Gill [2006] analysed natural language e-mails from people who had also taken a personality questionnaire. Part-of-speech and n-gram analysis showed distinct differences between personality type; for instance extroversion was correlated with frequent adjective use, while neuroticism was correlated with adverb usage. Ludwig et al. [2013] found that style affected online purchasing behaviours. Product reviews with a style matching the target market had a greater influence on purchasing decisions than those which failed to do so. More broadly, linguistic style is known to play a key role in communication accommodation theory [West and Turner, 2010], which governs many social interactions.

Communication accommodation theory [West and Turner, 2010] states that human communication behaviours change in response to an audience. There are two main processes that apply to style: convergence, where individuals attempt to match style; and divergence, where they seek to differentiate themselves. Convergence increases the effectiveness of communication, aids understanding, and is an indicator of the strength of a relationship [Pardo et al., 2012]. Divergence is used to position oneself as distinct, or to display membership of a particular social group. It is often used by professionals meeting with clients, but can be seen as a sign of dislike within a social setting [Ebesu Hubbard, 2009].

Much of the evidence for communication accommodation was collected through controlled experiments on face-to-face [Bilous and Krauss, 1988] or written [Niederhoffer and Pennebaker, 2002] communication. More recently, large scale experiments on social media platforms have validated communication accommodation theory [Danescu-Niculescu-Mizil et al., 2011; Doyle et al., 2016], and demonstrated the importance of accommodation in online discussions. In a controlled study of MBA students, Huffaker et al. [2011] demonstrates a positive correlation between linguistic convergence and agreement in multi-party negotiations. Working with the social-media platform twitter, Pavalanathan and Eisenstein [2015] show users modify their writing style for their audience. Posts written for small or geographically defined audiences tend to use more non-standard terms; for broader audiences more

generic terms are preferred. An analysis of tweets about the 2014 Scottish independence referendum by Shoemark et al. [2017b] showed a correlation between writing style and political stance. Distinctly Scottish terms were used more frequently by pro-independence authors. A later work [Shoemark et al., 2017a] showed audience and topic have relatively independent effects on style. Key to these studies is measuring linguistic style cohesion, frequently based on Linguistic Inquiry Word Count (LIWC) [Pennebaker et al., 2007] categories.

It is well known that linguistic style varies with geographic region, the study of which is called dialectology. These linguistic style variations are thought to have developed by a combination of isolation and the natural tendency of language to evolve over time [Johnstone, 1999]. However, in a world with global high speed communication, regional differences are taking on a symbolic value that marks inclusion in a social group. For example, adopting a regional dialect can help to make a sale [Johnstone, 1999]. Several authors have attempted to use linguistic features to predict an author's location [Eisenstein et al., 2010; Cheng et al., 2010].

2.5 Separating Style from Content

The separation of style from content is key to generating text in a particular style. In this section we explore the separation of style and content used in domains such as vision and speech, before examining techniques applicable to the text domain. For text, we focus primarily on authorship attribution because it is a well developed area of natural language processing that relies heavily on the ability to separate style from content.

There are a few different ways to separate content and style. The most basic, and the one most often used in text [Tausczik and Pennebaker, 2010], is to define components as wholly style or content. The classification of components can be done either manually or using a mixture of automatic and manual methods. Another approach is to use a dataset with style and content tags to train a factorised model that explains these two types of tags [Tenenbaum and Freeman, 2000; Popa et al., 2009]. Models without explicit factorisation have also proven effective for generation, but do not separate content and style [Van Den Oord et al., 2016; Gibiansky et al., 2017]. Alternatively, if only content tags are available, then training a deep content classifier [Gatys et al., 2016] or generator [Radford et al., 2017] can produce features that correlate with a particular style. Identification of these features is typically a manual process that relies on prior knowledge of the important style attributes (e.g. sentiment) expressed in the data.

In the vision domain, separating style from content has seen strong interest

from the literature and mainstream media. While early approaches to the problem achieved some success on human faces with bilinear models [Tenenbaum and Freeman, 2000], newer approaches [Gatys et al., 2016] using CNNs have produced visually striking examples of natural images in the style of famous paintings. To achieve this, first, a CNN is used to extract a content representation from the target image and a style representation from the painting. Next, a white noise image is passed to the same CNN and adjusted via back-propagation to match both the content and style representations. The content is encoded in CNN features (activations from convolutions) from each layer. The style is encoded in the dot product between CNN feature maps for each layer. Since layers represent different levels of specificity in terms of content and spatial extent, the relative weight given to content and style representations must be adjusted on a per-layer basis. Typically, larger weights are given to the style matching objective in earlier layers as these features in early layers tend to be less content specific. This method of style transfer is specific to images and cannot be readily applied to text style transfer.

In the speech domain, style separation can be used to adjust speech patterns to imitate the style of a different speaker – a task called voice conversion. An early method for voice conversion presented by Abe et al. [1988] learns a mapping between two speakers codebooks by counting correspondences in aligned training data. Although this does not explicitly separate style, the weights of the mapping from speaker A to speaker B can be seen as the style of speaker B expressed in terms of the style of speaker A. Popa et al. [2009] use a bilinear model [Tenenbaum and Freeman, 2000] to explicitly separate characteristics of the speaker’s voice from the spectral and phonetic content. Specifically they represent the styled speech vector y^{sc} by a style dependent weighting w^s of parameter vectors a^s and b^c . Where s is the style index and c is the content index.

$$y_k^{sc} = \sum_{i,j} w_{i,j,k}^s a_i^s b_j^c \quad (2.29)$$

Empirically, this gave better results than the previous gaussian mixture model approaches [Kain and Macon, 1998]. Recent approaches to generating speech with neural networks do not explicitly model the separation of style but do show the ability to generate speech in different styles [Van Den Oord et al., 2016; Gibiansky et al., 2017]. In these cases, discrete style attributes included in the ground truth, such as speaker ids, are input to the network.

Techniques for separating style and content in natural language often focus on classifying words. While this often leads to a more restricted view of style than the one taken by this thesis, many of our approaches have a basis in classifying words into different styles. Tausczik and Pennebaker [2010] describe natural language as

consisting of two broad categories of words: content words including nouns, verbs, and adjectives; and style words including prepositions, articles and conjunctions. They note that style words, otherwise known as function words, make up 55% of all words in natural language, despite constituting only 0.05% of the average person's vocabulary. The somewhat contentious [Bebout, 1993] content and function/style word categories do not map directly to the definitions used in this thesis, while we consider function words to be style words we also consider that words defined by [Tausczik and Pennebaker, 2010] as content words may in-fact carry significant style information. Nevertheless, narrowly defined style and content terms have shown application in the field of neurolinguistics to study patients with communication difficulties. In such contexts, the main difference between the categories appears to be their imageability (the ease with which a mental image is formed) [Bird et al., 2002] – with function words frequently being more abstract and less imageable. The syntactic distinction between content and style words provides a possible way of separating content and style; however, the distinction is not perfect and the non-syntactic imageability score [Coltheart, 1981] could also be considered.

2.5.1 Topic Models

Topic models, such as Latent Dirichlet Allocation (LDA) [Pritchard et al., 2000; Blei et al., 2003], are a powerful tool for automatically separating content from style in natural language. LDA is a hierarchical Bayesian model where: for each document a multinomial distribution over topics is chosen from a Dirichlet prior, then for each word in the document a topic is chosen from this multinomial, finally the word is drawn from a multinomial over the vocabulary for the chosen topic. An algorithm such as Expectation-Maximisation can be used to find the maximum a posteriori estimates for the parameters of the Dirichlet prior and the set of multinomials for each topic. However, as LDA is unsupervised its results are often difficult to interpret [McFarland et al., 2013], and the topics are not naturally aligned with style and content dimensions. Here we briefly summarise some extensions to LDA that directly relate to linguistic style.

Jin et al. [2011] develop a topic modelling approach based on Latent Dirichlet Allocation (LDA) that uses long texts to improve the modelling, and clustering, of short texts. Their model builds two separate distributions over topics, one tuned for the long texts and another tuned for the short texts. A binomial random variable decides, for each word, which topic distribution to draw from, while suitably chosen priors and constraints encourage the two topic distributions to be representative of their respective text sources: short or long texts. The separate topic distributions help

to model the different styles used in short and long texts, while the binomial switch variable allows the short texts to use topics primarily derived from the longer texts.

Titov and McDonald [2008] develop an LDA based topic model that models topics at two different levels of granularity: local topics and global topics. A local topic distribution is chosen for each window of T sentences around the current sentence, while the global topic distribution is chosen for each document as per traditional LDA. Intuitively, the local topics model text that is common across many documents but generally localised within each document, while the global topics model text that is typically different between documents. When applied to product reviews the global topics tend to capture brands and product types (eg iPod, Sony Walkman) while the local topics tend to capture rate-able qualities (eg sound quality, bass).

Brooke and Hirst [2013] attempt to separate aspects of six different styles (colloquial, literary, concrete, abstract, subjective, and objective) using a variant of LDA. They align each topic with a style by seeding the word distributions with known style words. To ensure these topics are actually aligned with these styles they consider the number of optimisation iterations as a hyper-parameter, since convergence might not occur until the topics have shifted away from the desired stylistic dimensions. The best performing model, trained primarily on blog and social media posts [Burton et al., 2009] to infer the style of held-out style words used binary word occurrences rather than counts, and required two iterations.

2.5.2 Authorship attribution

The properties of writing style have long been studied for the purposes of authorship attribution. The review of this literature provides important insights into separating style from content in a machine learning context. Moreover, the semantic term space presented in Chapter 6 takes inspiration from the features commonly used for authorship attribution.

The authorship attribution task involves identifying the authors of unknown documents given a set of documents with known authors. Of course, there are many variants to this problem, including: author verification (was the document written by a specified author), short-text authorship attribution (for example identifying the authors of e-mails or social-media posts) and fine-grained authorship attribution (matching parts of a long text to its authors); however, the general structure remains the same, identifying who wrote a document using text derived features.

Authorship attribution has a rich history stretching at least as far back as the 19th century [Mendenhall, 1887]. This review does not attempt to cover this vast array of work; instead, the focus is on modern statistical techniques for the English language

and the insights they give on the nature of style. Separating style from semantics is of primary concern in authorship attribution for establishing authorship, regardless of topic.

Authorship attribution is frequently cast as a supervised multi-class classification problem, where documents are classified as being authored by one of a fixed set of individuals. Features are extracted from each document and then used in an out-of-the-box classifier [Koppel et al., 2007] such as support vector machines or naive Bayes. It is these features that are most relevant to modelling the styled text.

2.5.2.1 Features

Key to many authorship attribution methods is engineering features that encompass style but not semantics, allowing author identification regardless of topic. Feature engineering is primarily a manual process involving considerable domain and linguistic knowledge [Stamatatos, 2009]. However, automatic methods are sometimes used for selecting important features from a large number of candidates. Recent deep learning methods – where the model generates new features from the raw data – have shown some promise [Ding et al., 2016].

Manual feature engineering has been a major focus of the literature, with thousands of different types of features being proposed [Stamatatos, 2009]. These features fall into four broad categories: character, lexical, syntactic and semantic features. Character features are the lowest-level, consisting of letter, digit and punctuation statistics. Lexical features encompass word or sentence level statistics such as n-gram frequency counts, common spelling errors or average sentence length. Syntactic features require a higher-level of natural language processing; they typically include POS tag counts, sentence and phrase structure, and grammatical errors. Semantic features typically involve statistics over synonym choices and semantic dependencies.

A number of authors [Argamon and Levitan, 2005] demonstrate remarkably high performance using only character and lexical features. In fact, Argamon and Levitan [2005] achieve high accuracy on a book identification corpus using only a bag of function words. Typically, the function words are chosen by experts to be predictive of style rather than semantics. This set varies with the text domain: for example, when working with newsgroup texts, Argamon et al. [2003] needed to use 190 Internet slang terms in addition to 303 generic function words. Van Halteren et al. [2005] hypothesise that texts by less experienced authors are harder to identify because they have a less distinctive style. To support this they show that additional token distribution features are required to classify texts from university students.

2.6 Styled Text Generation

The area of styled text generation aims to develop methods for generating text in a desired style, but is not focused on image-captions. Many styled text generation approaches do not control for content. Instead, they aim to capture the distribution of sentences under a particular style. Nonetheless, styled text generation techniques are closely related to styled image-caption generation. First, this section reviews a number of different styled text generation techniques, before considering poetry generation. Finally, two interesting techniques for styled text generation, variational auto-encoders (VAEs) and generative adversarial networks (GANs) are explained. This thesis uses neither VAEs nor GANs but they are prominent methods that require careful consideration.

Xu et al. [2012] use a collection of Shakespeare plays aligned with modern English to benchmark style translation approaches. Methods include phrase-based translation from the machine translation literature [Koehn et al., 2007], dictionary based translation (from human curated Shakespeare dictionaries), and unaligned translation where a phrase-based translator is learnt on a separate dataset but decoded with a language model trained on Shakespeare. They find that a standard phrase-based model outperforms all others except when evaluated on semantic accuracy, where the out-of-domain phrase-based model is superior. Furthermore, they show that compared to BLEU score, human evaluations are more correlated with a language model for target style and a linear classifier trained to distinguish styles.

Ficler and Goldberg [2017] generate movie review fragments conforming to a discreet set of 4 style (professionalism, length, descriptiveness, personal voice) and 2 content parameters (theme, movie rating score). Their results outperform an unconditioned model, and can generalise to unseen combinations of parameters. However, the degree of style and content control is coarse, limiting possible applications.

Oraby et al. [2017] consider the problem of generating stylistically interesting restaurant reviews for interactive dialogue systems. They learn sentence templates by collating dependency parsed restaurant reviews with dictionaries of subject types including: food, service and restaurant-type.

Jhamtani et al. [2017] translate modern English into Shakespearean English using an attentive RNN variant of pointer networks [Merity et al., 2016]. Because of the small quantity of sentence aligned data – only 16 Shakespeare plays aligned to modern English [Xu et al., 2012] – they made use of word-embeddings, pre-training and a dictionary of word level replacements encoded via pre-training.

There have been a number of attempts at automatically generating poetry [Gervás, 2001; Díaz-Agudo et al., 2002; Manurung, 2004; Wong et al., 2008b; Zhang and La-

pata, 2014; Qixin et al., 2016; Ghazvininejad et al., 2016]. This is a special case of language generation, as poems often conform to rigid grammar, rhythm or rhyming rules inherent to the poetic form eg Limerick, Haiku, or Sonnet [Manurung et al., 2000; Ghazvininejad et al., 2016]. Zhang and Lapata [2014] generate Chinese poetry with an RNN decoder conditioned on all previous lines, and with explicit decoding constraints. Previous lines are individually embedded by a CNN, while an RNN encoder merges line embeddings – an alternative is to use attention over all previously generated characters [Qixin et al., 2016]. Ghazvininejad et al. [2016] generate English language poems using a flexible finite state model that encodes domain knowledge. Specifically, an LSTM language model is decoded with rhythm, rhyming, and global structural constraints encoded into a finite state machine. Automatically generated poems can sometimes be controlled using context defined by a: word [Ghazvininejad et al., 2016], set [Zhang and Lapata, 2014], or sentence [Qixin et al., 2016], to which the resulting poems are loosely related. For poetry this is an acceptable, even necessary trope [Manurung et al., 2000], but in many other cases content control is important e.g. image captioning, sentence simplification and styled re-writing.

Variational auto encoders (VAE) [Kingma and Welling, 2013] for text [Bowman et al., 2016; Hu et al., 2017c; Yang et al., 2017b; Semeniuta et al., 2017] are sequence-to-sequence models that impose a prior on the latent embeddings; typically a multivariate Gaussian with zero mean and identity co-variance. The VAE loss function balances reconstruction loss with the KL-divergence of the latent space from the prior. The KL term encourages the latent embeddings to compactly fill the space around the origin and allows random text to be generated by sampling from the multivariate-Gaussian. Bowman et al. [2016] introduce the practical techniques of word dropout and KL cost annealing for training VAEs. They note that the latent space is relatively smooth: sampling from the space produces reasonable sentences, and interpolating between points in the space produces a sequence of sentences that interpolate between the two original sentences. Hu et al. [2017c] develop a VAE method for generating text with independent control of style attributes, such as sentiment and tense. Discriminators used during training ensure the generated text expresses the desired attributes. Current VAEs generate convincing sentences, but have weaker semantic content control because of the trade-off between reconstruction loss latent space constraints [Bowman et al., 2016; Semeniuta et al., 2017].

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014; Salimans et al., 2016; Arjovsky et al., 2017] are an emerging method for training text generators [Gulrajani et al., 2017; Yu et al., 2017; Li et al., 2017; Yang et al., 2017a]. A GAN can be thought of as a replacement for the common Kullback-Leibler (KL) divergence loss function between real data distribution P and generative distribution Q . It is an ap-

proximation to the Jensen-Shannon divergence: a loss which interpolates between forward $KL[P||Q]$ and reverse $KL[Q||P]$ divergence [Goodfellow et al., 2014]. GANs consist of a generator network, for generating text, and a discriminator network which attempts to classify text as either real (human generated) or fake (network generated). In an auto-encoding model classification in the latent feature space is also possible [Hu et al., 2017b]. Training is typically iterative, switching between training the generator to fool the discriminator and training the discriminator to identify the current generator’s output. This requires computing gradients of the discriminator’s loss function with respect to the parameters of the generator – a non-trivial task in a discrete space such as text. One solution [Gulrajani et al., 2017; Press et al., 2017; Hu et al., 2017b] is to feed the generators output probability distribution directly into the discriminator, and only sample discrete tokens at test time. This ensures differentiability, but means there is a large difference between how the model is trained and how it is used. To overcome this, a policy gradient method may be used to train the generator [Yu et al., 2017; Li et al., 2017; Yang et al., 2017a], with partial sequence rewards achieved by applying Monte Carlo search [Yu et al., 2017; Yang et al., 2017a] or training a separate discriminator [Li et al., 2017]. GANs have not yet seen widespread adoption for text generation since they are still relatively difficult to train when the output space is discrete. Moreover, maximum likelihood methods for training text generators work well.

2.7 Style Transfer for Text

A number of approaches to style transfer for natural language text have been released concurrent with, or post, the work which constitutes this thesis. These approaches fall into two main groups, those that use an adversarial loss and those that need sentence aligned training pairs to separate content and style. There are also a number of approaches that do not fall into these categories, instead using back-translation [Prabhumoye et al., 2018], word frequency [Li et al., 2018], or parameter sharing [Han et al., 2018] to separate style and content.

2.7.1 Adversarial Approaches

Shen et al. [2017] present two auto-encoding models for text style-transfer that work without needing sentences to be aligned between the two styles. Both models utilise a separate encoder and decoder for each style domain, which transfer to and from latent space shared between the two styles. Their first model, the aligned auto-encoder learns the shared space by incorporating, into the loss function, an adversarial dis-

criminator that attempts to classify latent representations into their original style. This ensures that the latent representation encodes only sentence properties that are shared by both styles. Their second model, the cross-aligned auto-encoder also uses an adversarial discriminator applied to latent states, but in this case a discriminator is applied to each hidden state of the generator. Using human evaluation and automatic classifier evaluation they show the applicability of these models to sentiment modification, word substitution, and word order recovery.

Fu et al. [2018] build style transfer models for text that use an encoder-decoder neural network structure with an adversarial classifier in the latent space. The encoder is shared across different styles. In their first model a separate decoder is learnt for each style, in their second model a single decoder is used with a learn style vector concatenated to the final hidden state of the encoder. They apply their model to paper title to news title style transfer and positive to negative review style transfer. Human evaluations measured content preservation while automatic classifiers measured style transfer strength.

Santos et al. [2018] build an encoder-decoder model for style transfer using a combination of reconstruction loss and style classification loss. The style classifier is trained along with the model: functioning as an auxiliary adversarial loss. In addition their model has a separate backwards transfer step, where the discrete style-transferred output words are fed back into the encoder-decoder with the goal of optimising reconstruction of the original sentence and classification loss. This backwards transfer step is separate as the discrete outputs of the encoder-decoder cannot be back-propagated. Evaluation using an automatic style classifier and content preservation metric (based on mean GloVe embedding) shows that their model outperforms the cross-aligned auto-encoder of Shen et al. [2017] in the task of translating offensive sentences into non-offensive variants.

2.7.2 Approaches Requiring Sentence Alignment

Carlson et al. [2017] build a style transfer system inspired by recent work on multilingual translation [Johnson et al., 2017]. Their system requires sentences aligned across many different styles, in this case bible versions. An RNN encoder-decoder model is shared across all styles, with a special indicator token appended to the input to specifying the desired output style. They do not explicitly try to remove style in the latent space between encoder and decoder, instead relying on the large number of different styles and a relatively small intermediate space to ensure the latent space is compact and discards style information that doesn't come from the easily accessible indicator token. For evaluation they use BLEU and PINC which

computes the fraction of n-grams in the generated sentence that are not in the original sentence.

Rao and Tetreault [2018] benchmark a number of machine translation techniques for style translation using a crowd-sourced dataset of paired formal and informal sentences. They find that a traditional probabilistic machine translation approach, based on Moses, performs the best; however, it is not significantly better than a neural machine translation approach adapted for style translation [Jhamtani et al., 2017]. For the neural machine translation approach they extended the number of training sentences using the Moses baseline – perhaps leading to the similar performance of the models.

2.7.3 Other Approaches

Prabhumoye et al. [2018] approach the task of building a style independent representation of text using back-translation. Specifically, they train two encoder-decoder translation models, one from English to French, another from French to English. The style independent representation is formed by translating the English into french and then using the encoder portion of the French to English translator. By fixing the parameters of the two translation models they then train decoders in the desired style using a loss function that is a weighted sum of reconstruction loss and likelihood under a pre-trained style classifiers. A continuous approximation to the softmax is used to allow gradients to be back-propagated through the decoder output layer. For some, but not all, style domains (gender, political slant, and sentiment) their model performs better than the cross-aligned auto-encoder of Shen et al. [2017] in classifier evaluations of style transfer and human evaluations of meaning preservation.

Li et al. [2018] use a deletion, retrieval, generation process to create sentences that have a particular style attribute, such as positive or negative sentiment. From an input sentence they delete n-grams that frequently occur for a particular style, relative to other styles, and then attempt to fill in the blanks by retrieving similar sentences in the target style. Their best performing model, is an encoder-decoder RNN, with two separate encoders, one for the input sentence and another for the retrieved sentences. It is observed that this model has a strong inductive bias towards target style attributes that are likely to fit the context of the input sentence, when compared to an encoder-decoder baseline without retrieval.

Han et al. [2018] build an encoder-decoder model with two style switches, one between the word embedding layer and the encoder RNN, and the other between the output fully connected layer and the decoder RNN. This switch can be seen as a way of separating the input and output weight matrices of the RNN into independent

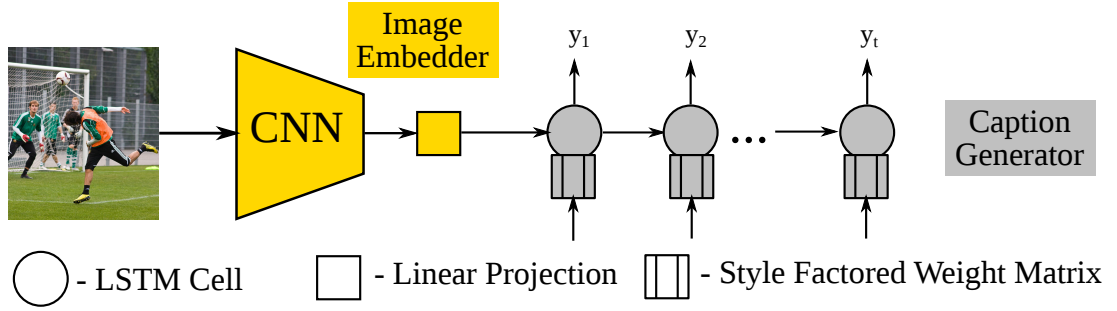


Figure 2.2: The StyleNet model [Gan et al., 2017a] for generating styled captions.

parts for each style. The model does not require paired sentences for training.

2.8 Caption Generation with Style

Only a few authors have considered the problem of generating image-captions with style. The two most relevant systems are StyleNet, proposed by Gan et al. [2017a], and neural-storyteller, proposed by Kiros [2015]. However, a number methods for generating image captions with sentiment which build on our SentiCap system (developed in Chapter 4) exist. This section reviews StyleNet, neural-storyteller, sentiment captioning methods, and briefly touches on the related problem of multilingual image captioning. The purpose is twofold: to show how our work fits within the literature on styled image-caption generation, and to explore the research that our SentiCap work has inspired.

StyleNet [Gan et al., 2017a], shown in Figure 2.2, is a recent approach for generating image captions with a particular style. It uses a CNN+RNN with an LSTM unit where the weight matrix W_x applied to the input word embedding is factored:

$$W_x = U_x S_x V_x \quad (2.30)$$

With style component S_x and shared components U_x, V_x . Hidden state transition matrices are not factored. The CNN+RNN is trained first end-to-end on the Flickr30k dataset and then the style components S_x are trained in a language model set-up – a random semantic vector is provided to the LSTM in the first time step. The resulting captions are weakly styled and semantically relevant; however, the training uses crowd sourced styled image-captions which are time consuming and expensive to collect.

neural-storyteller [Kiros, 2015], shown in Figure 2.3, is a system for generating short styled stories about images without an alignment between images and styled

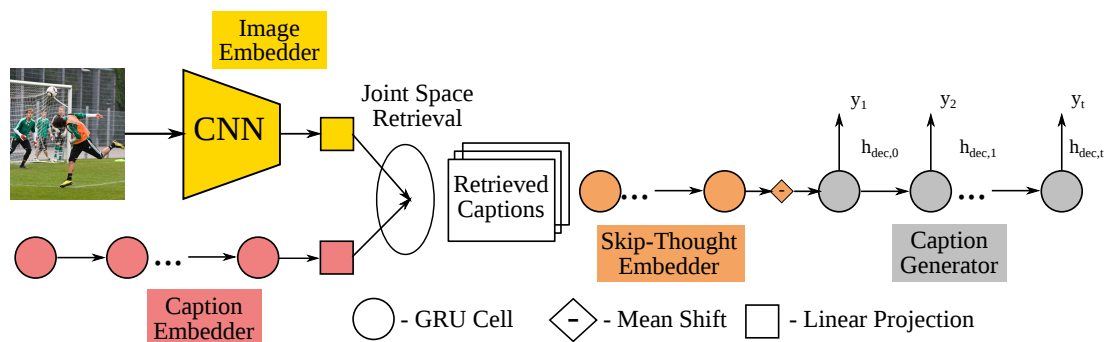


Figure 2.3: The neural-storyteller model [Kiros, 2015], for generating short styled stories about images. The mean shift block subtracts off the mean skip-thought vector for captions and adds on the mean skip-thought vector for the target style.

captions. It relies on a semantic sentence embedding technique called skip-thought [Kiros et al., 2015]: RNN sentence embeddings trained to predict surrounding sentences. An image is projected into a vector space [Kiros et al., 2014], where similar captions are retrieved and then projected into thought-vector space. Style shifting is performed by subtracting off the mean skip-thought vector for captions, and adding the mean skip-thought vector of the target style. This style shifted vector is decoded by a conditional RNN language model trained on the target style. The resulting captions are clearly representative of the target style, but are only loosely related to the image – in general the semantics are lost.

2.8.1 Captions with Sentiment

The following research was released after publication of the SentiCap method in Chapter 4. They provide a number of new solutions to generating captions with sentiment. We briefly discuss the pros and cons of these new solutions.

Karayil et al. [2016] develop a model that generates image captions with strong sentiment. The three components of this model are visual concept detection, graph based sentence generation, and template filling. Visual concepts are detected by thresholding Adjective-Noun-Pair (ANP) detector scores obtained with *DeepSentiBank* [Chen et al., 2014]. The *DeepSentiBank* ANPs have strong sentiment polarity (eg “happy dog”, “scary dog”). Sentence generation uses a graph where the nodes are common content words (eg adjectives, nouns, verbs) and the edges are strings of function words (eg prepositions, conjunctions, pronouns). Edge weights are calculated by counting the number of occurrences in the training set. Additional zero weighted edges are added between nodes with a similar word2vec [Mikolov et al., 2013] score. The final sentence is generated by finding a path with high total edge weight, passing through

all activated ANPs. If no such path is found the fall-back is simple template filling of “*HUMAN with PROPERTY doing VERB on EVENT in LOCATION.*” – partial fillings are also considered. The entire model is trained on image-captions sourced from social media and thus avoiding costly or time consuming image annotation. However, when evaluated on the manually constructed MSCOCO caption dataset, they achieved a BLEU score an order of magnitude weaker than the state-of-the-art. Differences between the training set and MSCOCO likely contributed to this low score, as did the focus on adjective noun pairs with strong sentiment. It would be interesting to evaluate the graph based approach against RNNs and template filling without the domain shift between training and testing.

Andrew Shin and Harada [2016] use a CNN+RNN framework to generate captions with sentiment for images. Rather than using a single CNN trained on ImageNet, they train an additional CNN on sentiment terms. Training images tagged with sentiment terms are retrieved from online image hosting services. A CNN model originally trained on ImageNet is then fine-tuned on this dataset in a multi-label learning setting. The RNN is trained on descriptive datasets such as MSCOCO or Flickr 30k, using features from both the ImageNet CNN and the sentiment CNN. At test time, they force the RNN to output a sentiment word right before the most likely noun. To do this the entire sentence must be generated and hidden states recorded. Then a single RNN cell is run using the input to the cell that generated the most likely noun, only with a vocabulary restricted to sentiment words. The resulting captions are descriptive of the image and introduce sentiment that is more appropriate than compared methods. They do not compare with SentiCap (Chapter 4), perhaps because the task is slightly different: they try to determine a sentiment for the image before captioning, while SentiCap allows the user to choose the sentiment. Their method works by encouraging the use of sentiment words already existing in the descriptive dataset. Merging in a dataset which frequently uses sentiment words in captions such as the one presented in Chapter 4 may improve performance. Likewise, exploiting unary word probabilities from their fine tuned sentiment CNN during decoding could prove beneficial.

You et al. [2018] introduce some modifications to a CNN+RNN framework to enable positive and negative sentiment generation. The first approach appends a sentiment unit to all input word embeddings and adds a word level sentiment loss. The sentiment unit takes a scalar value of $\{-1, 0, 1\}$ (corresponding to negative, neutral and positive sentiment) set to the sentiment of the ground truth sentence during training and the desired sentiment during generation. The word level sentiment loss is an auxiliary loss on the log probability of the ground truth sentiment – computed by a multilayer perceptron applied to the output of the RNN. The second approach is

to add an additional memory cell to the LSTM RNN, with hidden state initialised by an embedding of the sentiment value: negative, neutral, and positive. An auxiliary loss is also used, but only applied to the last step of the sequence, rather than at every step. This auxiliary loss helps ensure the network remembers the sentiment polarity. Both approaches are trained using the MSCOCO dataset and the SentiCap dataset (which as a product of this thesis is detailed in Chapter 4). In automatic evaluations, both their methods outperform SentiCap presented in Chapter 4. However, it remains to be seen which factors contributed most to the overall quality of the sentiment captions: the improved visual features from ResNet-152 (vs VGG-16 for SentiCap), or one of the new loss components.

2.8.2 Multilingual Captioning

Multilingual image captioning involves generating captions in different languages or using multiple language resources to improve captioning performance. Although related to stylistic image captioning, the multilingual task requires very different outputs for each language, with almost every word changed. CNN+RNN models are a common approach, for example Jaffe [2017] evaluates a number of different RNN configurations on English and German. Their best performing model is a CNN+RNN for German, where the last output layer of the German decoder is input to an English decoder, with the whole structure trained jointly using English and German ground-truth. The WMT 2016 shared task [Specia et al., 2016] involved generating German translations of existing English captions. The best submissions were based on off-the-shelf phrase-translation tools [Koehn et al., 2007] with re-ranking based on visual features. This stands as a testament to the quality of modern phrase-translation tools, and positions them as a possible baseline for style generation. In the WMT 2017 shared task [Elliott et al., 2017], the best ranked system was a CNN+RNN model with attention over both the input image and the source sentence. WMT'17 also included a task to generate German directly from the image, without an English source caption; no system beat the baseline [Xu et al., 2015a] trained only on German. This demonstrates the rapid improvement in CNN+RNN models for multilingual captioning and the importance of aligned image-caption data in the target domain.

2.9 Summary

This thesis builds on a large base of work in computer vision and natural language processing. In particular I build on recent advances in object detection [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016],

image captioning [Donahue et al., 2015; Karpathy et al., 2014a; Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015b], and natural language generation [Mikolov et al., 2010; Graves, 2013; Cho et al., 2014a]. I use ideas from authorship attribution to represent image-caption content independent from style, and adapt sequence-to-sequence models previously used for machine translation. The overarching aim is most similar to StyleNet [Gan et al., 2017a] and neural-storyteller [Kiros, 2015]. While research that is part of this thesis has also influenced similar works [Karayil et al., 2016; Andrew Shin and Harada, 2016; You et al., 2018]. As each individual chapter of this thesis focuses on a slightly different aspect of styled image-captioning, the specifically relevant literature is reviewed in the related work sections for each chapter.

Object Naming for Image Captions

3.1 Introduction

Object names are an essential part of image captions, communicating much of the semantic information, and representing a stylistic choice [Simpson, 2004]. In fact, categorisation and naming is central to our interpretation of the physical world [Lakoff, 1987]. Understanding and predicting naming choices is therefore an important goal for stylistic image captioning. In this chapter I develop name selection techniques for visual concepts, and analyse naming patterns across a huge number of concepts. For name selection, I focus on exploiting visual context and using image-caption pairs mined from the web to avoid expensive data annotation. The analysis of naming patterns focuses on the degree of naming variability ‘in the wild’ for both concepts and sub-trees defined in a naming hierarchy. In analysing these, I reveal the degree of stylistic freedom in concept naming and develop a method for choosing names that is applicable to styled caption generation.

A single concept may have multiple names: consider the concept “*gala apple*” (a popular type of apple), which could equally be named as “*apple*” or “*fruit*”. In many situations, people consistently choose the same name, called the basic-level [Rosch et al., 1976]. In this case the basic-level name is “*apple*”. We can view the basic-level name as the default choice: if you have no other information, you should use the basic-level name. However, visual concepts often come with contextual information that can consistently change the names people use [Lakoff, 1987; Rosch, 1999]. For example, in the presence of other fruit, the “*gala apple*” may be collectively described as “*fruit*”. Alternatively, if the apple is up for sale then the specific name “*gala apple*” may be more appropriate. In this chapter I develop an automatic concept naming method that takes visual context into account.

Automatically naming objects in an image is one of the most ambitious tasks of computerised image understanding. Progress on this task alone has important implications, with billions of pictures on the web and in personal or professional

collections. There are two aspects to this image-to-name problem. The first is visual concept recognition for thousands of visual semantic categories – otherwise known as object recognition, reviewed in Section 2.1. Systems that solve this problem have made remarkable progress [Krizhevsky et al., 2012; Felzenszwalb et al., 2010]. The second aspect is to mimic the human description of categories – recent work by Ordonez et al. [2013] addressed this aspect by identifying basic-level names. There are two key limitations of recent literature on the image-to-name problem. The first is assuming the basic-level name for a visual category is unique, whereas cognitive psychology acknowledges that object naming is context-dependent [Barsalou, 1982; Mareschal and Tan, 2007; Chaigneau et al., 2009] and affected by attributes such as typicality [Jolicoeur et al., 1984]. The second is the reliance on explicit crowd-sourced labelling to map from categories to basic-level names. While crowd-sourcing is an efficient way to gather one name-per category for tens of thousands of categories, it does not scale to the context dependent case. In this chapter, I scale image-to-name systems to millions of images to study the interplay between names and context – both visual and language. To this end, I make use of millions of online images with human-supplied descriptions [Deng et al., 2009; Xie and He, 2013], and large-scale visual recognition systems [Jia et al., 2014; Krizhevsky et al., 2012].

This chapter has two classes of contribution: new methodology for selecting names, and an analysis of naming choices. Sections 3.3 & 3.4 present both new methodology and analysis, while Section 3.5 focuses on analysis. In Section 3.3 I examine naming patterns in the MSCOCO dataset, where both object ground truth and human-generated natural language descriptions are available. This section develops new methodology for choosing names for known concepts based on visual context. It also validates context as a factor in naming and explores the types of concepts most affected by visual context. In Section 3.4 I expand the contextual naming method to a web-scale image-caption dataset with automatic visual concept detectors. I observe that for many concepts with more than one frequent name, context is a strong predictor of the chosen name. In Section 3.5, I focus purely on analysing name usage across hierarchical concept structures through an exploration of animal naming in the Linnaean hierarchy.

3.2 Background

The study of naming and categorisation traditionally belongs to the discipline of psychology. One key concept is the basic-level [Rosch et al., 1976] name: an abstraction level appropriate for most contexts in which the object appears. Basic-level names also tend to be used frequently in day-to-day interactions, and are relatively short,

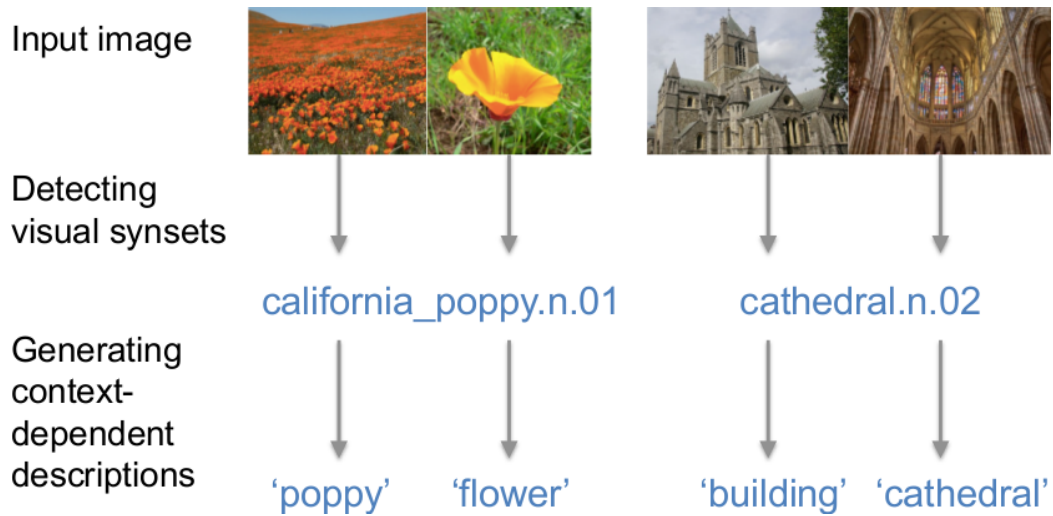


Figure 3.1: Different names can be given to the same concept given a different view-point.

with few characters or syllables.

Basic-level names are a useful approximation, but there are more complex factors that people appeal to when making naming choices. Rosch [Rosch, 1999] notes that context affects naming, both in the level of abstraction used and even the choice of objects to name. Similarly Chaigneau et al. [2009] demonstrate a distinct change in naming after a situational information change. For example, knowledge of how a previously unfamiliar object fits into a known system, in this case a catapult system, changes the way subjects name objects as either fulcrums, levers, weights projectiles or targets. The effect of context on categorisation has even been identified in infants before they have a full grasp of language [Mareschal and Tan, 2007]. The diversity of toys initially given to infants significantly affected how they categorised a subsequent set of toys.

Visual appearance defines another set of factors affecting naming. Psychologists have identified some of these factors as typicality [Jolicoeur et al., 1984], perceptual variability, familiarity [Snodgrass and Vanderwart, 1980], and kind diversity [Mareschal and Tan, 2007]. In images, people tend to assign different names to different view points. The occurrence of other objects falling into the same semantic class also affects naming. Psychologists have observed a similar effect, namely context-independent and context-dependent properties [Barsalou, 1982]. For example, an entire field of *california poppy* flowers is described with the word *poppy*, while a single flower is often described with the word *flower* (Figure 3.1 left); given an outside photo a *cathedral* is likely to be named as a *building*, while inside it is more

commonly a *cathedral* (Figure 3.1 right, second row).

Automatically associating images with natural language is a very active topic within computer vision. Most recent systems rely on visual recognition as a component, such as state-of-the-art approaches using convolutional neural networks (CNN) [Krizhevsky et al., 2012; Jia et al., 2014] – see Section 2.1.2. The approaches for associating images to words and sentences started with visual detection over a small number of object categories, followed by language modelling [Yang et al., 2011], caption retrieval [Ordonez et al., 2011], and explicitly capturing syntactic and semantic features [Hodosh et al., 2013]. A few approaches explicitly relate visual semantics to their expression in words, such as studying how objects, attributes and visual relations correlate with their descriptions [Zitnick and Parikh, 2013], and learning visual variations of object categories [Divvala et al., 2014]. In terms of capturing human descriptions of natural images, our work is inspired by the studies of importance [Spain and Perona, 2011; Berg et al., 2012] and the first work to identify basic-level categories from images [Ordonez et al., 2013].

Part of our work is to explicitly model whether a concept is described in an image caption. This simulates the process of writing captions, where the author chooses, either consciously or subconsciously, if a concept will be named. Many visual concepts, particularly those that define the setting, may not be mentioned in the caption because they are not thought to be the focus of the image. This is an instance of figure-ground perceptual grouping [Wagemans et al., 2012], where the figures are the concepts mentioned in the caption while the background consists of concepts that are not mentioned. This differs slightly from other instances of figure-ground perceptual grouping. For example, Spain and Perona [2011] examine what objects in natural images are foreground by asking people to annotate 10 objects in the scene. This is different to our case since the objects people annotate do not necessarily relate one-to-one with those named in captions. Whereas, Berg et al. [2012] use a definition of figure-ground that mirrors our own, and find that context is a strong predictor of importance, though features such as size and concept type are also strongly correlated with importance. The figure-ground grouping also applies within captions themselves, with some concepts being highlighted as the figure and others forming the background [Talmy, 1975]. For example, given the sentence “*The car passing in front of a building.*” the figure is “*The car*” while the background is “*a building*”. This grouping may change how concepts are described, for example, if “*the building*” was the figure then the caption could be “*A church with traffic in front.*”. In this work we do not explicitly model the effect of focus on naming choice, though choosing names with the aid of contextual features does capture some focus dependent naming behaviour.

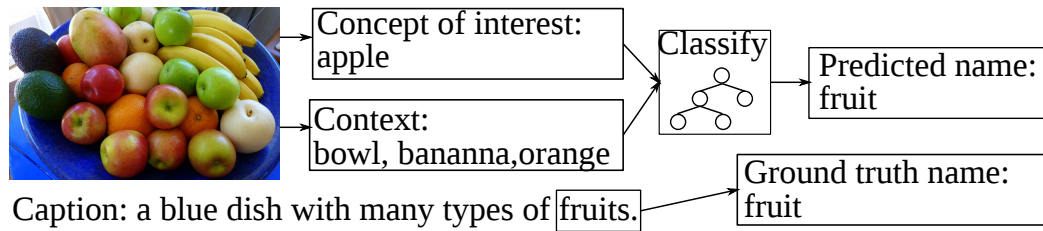


Figure 3.2: We build a classifier that predicts a name for a concept of interest. Ground-truth co-occurring objects are the features, while the caption defines the ground-truth name.

Our system for choosing names offers two points of departure from the state-of-the-art. First, the goal of deciding which names describe an image is different from detecting visual objects; the focus here is the choice of description rather than visual detection. This choice can be seen as stylistic rather than semantic in nature. Second, a number of recent works apply context to image understanding tasks [Divvala et al., 2014; Spain and Perona, 2011; Berg et al., 2012] or try to predict naming choices [Ordonez et al., 2013], but these two parts have not yet been connected. By combining them we are able to choose names more accurately.

Our analysis of image naming also departs from the current state-of-the-art. We explore the effect of context on thousands of concepts in situ, using images with captions generated naturally by hundreds of thousands of users. This is a major departure from typical cognitive psychology experiments that employ a few dozen to a hundred subjects to name or categorise a dozen isolated concepts [Rosch et al., 1976] represented by toys or drawings [Jolicoeur et al., 1984; Mareschal and Tan, 2007].

3.3 Object Naming in Context: A Pilot Study

In this section we begin to tackle the two main aims of this chapter: to develop a method for choosing names for images using context, and to analyse patterns in naming choices. First, given a visual concept and the context in which it occurs, we aim to predict the name used in the caption. In doing so, we can analyse the importance of visual context for choosing names. This pilot study avoids the uncertainty of using image-derived features, instead relying on a large image dataset with manual annotations. By restricting our context to object co-occurrence defined by ground truth image annotations, we can study visual context without using automatic image-derived features.

We use the Microsoft Common Objects in Context (MSCOCO) [Lin et al., 2014]

dataset with 82783 images, 413915 captions and ground truth object detections for 80 classes. Target names are extracted from image captions while the ground-truth object annotations form the contextual features, see Figure 3.2. Our method considers the name frequency for each object, and the impact of context on naming accuracy. Using decision tree classifiers, we both predict names accurately and show concrete interpretable cases where context affects naming.

3.3.1 Dataset and Pre-processing

The MSCOCO [Chen et al., 2015] training set has 82783 images, each with five captions collected from crowd-sourcing platform Amazon Mechanical Turk. Also available are manual annotations identifying which of 80 pre-defined MSCOCO concepts are present in each image. These annotations were collected independently from the captions and are designed to be complete: if a concept is in the image and visually recognisable it should be annotated.

We define object concepts using WordNet [Miller, 1995] noun synsets: groups of words with the same meaning. Noun synsets are arranged hierarchically: ancestors (called hypernyms) are more general terms, while descendants (called hyponyms) are more specific terms. For example the word *cat* in the sense of a feline mammal has a direct hypernym *feline* and a direct hyponym *domestic cat*.

We manually matched the 80 MSCOCO concepts to unique WordNet synsets; when paired with the existing annotations for each image this forms our *concept* ground truth. The *naming* ground truth comes from parsing the five captions for each image. The nouns are identified using a parts of speech tagger, then uni-grams and bi-grams are formed from the surrounding words. Each of these n-grams is then matched to the WordNet hierarchy and filtered down to synsets that are ancestors or descendants of the image’s ground truth concepts. We reduce overlapping n-grams by choosing the most specific, defined as the match to the synset in WordNet furthest from the root. This ensures that the bi-gram *tennis racquet* is matched to the *tennis racquet* synset, rather than the *racquet* synset, while the uni-gram *racquet*, occurring alone, is matched to the *racquet* synset.

3.3.2 Model

First, we define a set of visual concepts \mathbb{C} and a vocabulary of names for each concept $\mathbb{V}^c, c \in \mathbb{C}$ – see Section 3.3.1. The set of concepts represented by an image i is C^i . Our goal is to predict the name $y^{c,i} \in \mathbb{V}^c$ for each concept instance, given the set of

all concepts in the image. This is expressed as:

$$\underset{y^{c,i}}{\operatorname{argmax}} P(y^{c,i} | C^i) \quad (3.1)$$

We cast the problem as a set of independent multinomial classifications, with one classification task per concept. The set of concepts in an image C^i is represented by a multi-hot vector. Since there are only 80 classes in MSCOCO this is sufficiently compact.

First, we identify the concepts with multiple frequent names used in captions. A concept has multiple frequent names if there exists a name with at least 10% the frequency of the most frequent name. For these concepts we learn random forest classifiers to predict the name $y^{c,i}$, given the concept ground-truths C^i – we refer to this method as the *context-name* method. This model is compared against a *frequent-name* baseline, which always assigns the most common name for each concept. For example, providing this baseline with images marked as *bicycle* in the *concept* ground truth could lead to invariable prediction of the name *bike* if it is the most frequently used in the *naming* ground truth.

Random forest is preferred over other classifiers for its interpretability. To reveal the most important contextual objects for naming we use the Gini importance metric [Archer and Kimes, 2008], which has been used extensively by other authors to interpret the importance of features in decision forests [Louppe et al., 2013]. Intuitively, Gini importance measures how cleanly a feature divides the training data into the target classes. It is calculated by averaging the decrease in the Gini impurity (Equation 3.2) across all uses of a feature. A feature is said to be used when it is the decision variable for a sub-tree in the random decision forest. In order to define the Gini impurity for a sub-tree, we denote p_{y^c} as the fraction of training examples assigned to the current sub-tree, with ground truth name y^c . Gini impurity I_G (Equation 3.2) is then the probability of making a classification mistake for a randomly sampled example from this sub-tree.

$$I_G = \sum_{y^c \in \mathbb{V}^c} p_{y^c} (1 - p_{y^c}) \quad (3.2)$$

3.3.3 Learning Set-up

To learn the *context-name* model, we train on 80% of the pre-defined MSCOCO training set. A further 10% is used for selecting hyper-parameters and the final 10% is used for testing. Each random forest classifier consists of 100 trees, while the minimum number of examples for splitting an internal node is 4 and the minimum number of examples per leaf is 2.

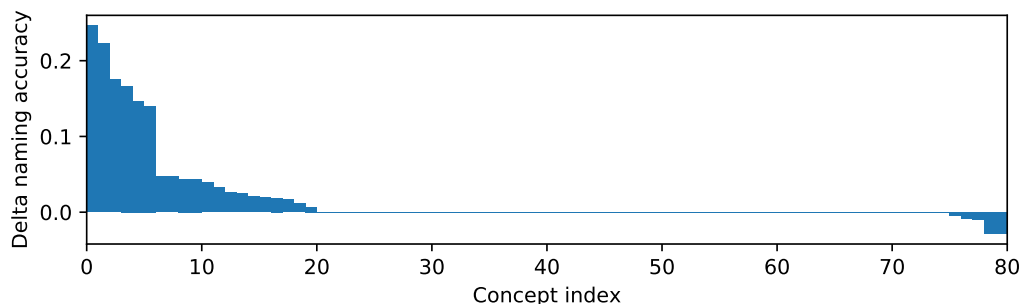


Figure 3.3: Improvement in name prediction accuracy when context is given compared to no context. All 80 MSCOCO concepts are shown ordered by improvement. See Fig 3.3 for raw accuracy scores.

We apply the random forest classifier in the sklearn [Pedregosa et al., 2011] library, the part-of-speech tagger from the spaCy¹ library, and the WordNet [Miller, 1995] lemmatizer from nltk [Bird et al., 2009].

3.3.4 Results

Of the 80 MSCOCO concepts, 48 have at least two common names, indicating that a single basic-level name is not an appropriate simplification for 60% of the concepts.

A comparison of the *context-name* method to the *frequent-name* method is given by Figure 3.3. Out of the 48 concepts with at least two common names, 9 showed an improvement in naming accuracy of greater than 5%. Figure 3.4 shows that without context these concepts are not accurately named although, even with context, they still, on average, exhibit slightly weaker naming accuracy than other concepts. In fact these concepts are among the hardest to name, with multiple names in common usage. Many of the concepts with no improvement are already named accurately because of highly skewed name frequency distributions. There are five concepts where the *frequent-name* baseline outperforms the *context-name* method. These concepts are characterised by relatively small testing and training sets, suggesting over-fitting is a likely cause.

The most-improved concepts are *car*, *ball*, *orange* and *backpack*. For *orange* the most common names are *fruit*, *oranges*, *food*, while the most important object context, as measured by the Gini metric, are *apple*, *dining table* and *bowl*. Intuitively, when people name an orange they are more likely to use the collective term *fruit* in the presence of other fruit such as apples. The concept *bowl* likely indicates that there is a fruit bowl in the image – which further supports this idea. In the case of *ball* the most common

¹<https://github.com/explosion/spaCy/tree/v1.9.0>

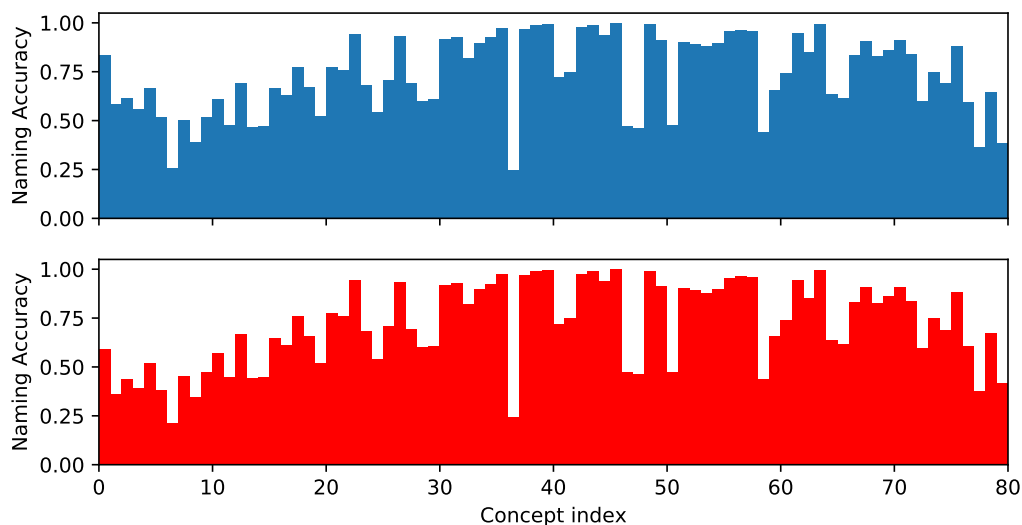


Figure 3.4: Name prediction accuracy when context is given (top) compared to no context (bottom). All 80 MSCOCO concepts are shown ordered by delta accuracy improvement from including context. See Fig 3.3 for delta improvements.

names are *tennis ball*, *baseball* and *ball* while the most important concepts are *tennis racket*, *baseball bat* and *baseball glove*. This is a case where the concept has multiple sub-concepts each with their own basic-level name. The context allows differentiation of the sub-concepts, which ultimately aids name selection.

This pilot study on annotated MSCOCO dataset shows many objects have context dependent names. Moreover, the co-occurrence aspect of visual context plays an important role in naming, allowing more accurate predictions of names used in captions. This opens the door to large scale naming, which I explore in the following sections.

3.4 Large Scale Object Naming with Visual Context

We propose a three step approach to automatically name objects in images. We first use ImageNet [Deng et al., 2009] to learn visual concept models for more than 2,000 visual synsets (Section 3.4.1.1). We then learn a model to name detected concepts, taking into account: visual context, object importance, and object appearance (Section 3.4.1.3). Finally, in Section 3.4.1.4 we rank the names on a per-image basis using a trained model that incorporates language context. This system is evaluated on the SBU 1 Million Flickr image dataset (Section 3.4.3). Our system achieves a higher precision and recall than frequency based basic-level naming [Ordonez et al., 2013]

on a top 5 prediction task. For 1,200+ visual synsets we see an improvement when incorporating visual context into the name prediction task, ultimately causing more accurate matches with human descriptions.

The main contributions of this section are:

- The large-scale verification of visual context as an important factor in object naming.
- The first large-scale catalogue of context dependent names for thousands of categories. This is automatically constructed by analysing web-scale datasets with natural language descriptions, and can easily scale to an order of magnitude more concepts.
- A new method for predicting context-dependent names taking into account visual and linguistic information. This is achieved by decomposing the problem into a set of classification and ranking tasks.
- Benchmarking on a dataset two orders of magnitude larger than prior work [Ordonez et al., 2013] shows our context-dependent naming system substantially improves word selection accuracy for the image-to-word task.

We have released our catalogue of context-dependent basic-level categories, and a word prediction benchmark of 150,000 images online ².

3.4.1 Method

We propose a method, represented in Figure 3.5, to predict the names of objects given an image. Our approach models the probability of using a name y to describe an image with feature vector \mathbf{x} , as $p(y|\mathbf{x})$. To accurately model this distribution we consider: the relationships between visual concept and images, the relationship between concepts and names, and the relationship between different names. This gives rise to three components: detecting visual concepts in images (Section 3.4.1.1), naming visual concepts (Section 3.4.1.3), and ranking names for all detected concepts using high level contextual image and co-occurrence features (Section 3.4.1.4).

3.4.1.1 Detecting Visual Concepts

The first steps towards deciding on descriptive names for an image are defining the visual concepts, and then recognising these concepts in the images.

²<https://github.com/computationalmedia/naming-with-visual-context>

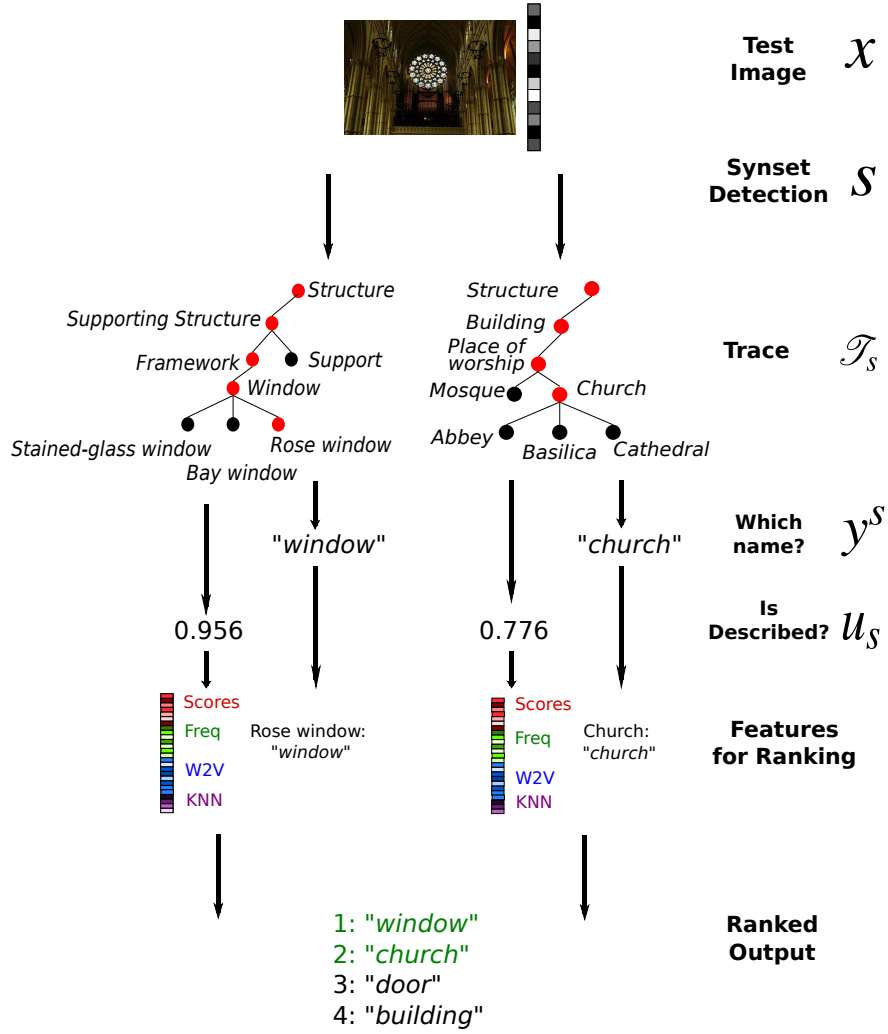


Figure 3.5: Method overview for context-dependent name prediction. See Section 3.4.1 for details.

We define our visual concepts using WordNet. This widely-used lexical database defines *synsets* that represent unique word senses [Miller, 1995]. In linguistics, a word sense is “an element from a given set of meanings” [Miller, 1995]. WordNet synsets have previously been used to define visual concepts for the well-known ImageNet [Deng et al., 2009] database, where each visually distinct WordNet synset is illustrated with a few hundred images. By defining our visual concepts on WordNet we can learn a visual representation for each synset with ImageNet.

Our concept detection model uses binary logistic regression classifiers with feature vector x extracted from the second last fully-connected layer of the BVLC Reference CaffeNet [Jia et al., 2014] CNN. Equation 3.3 adapts the second last layer of

a CNN to new synsets. In Section 3.4.1.3 we also use these CNN features for the contextual naming task. Training a simple classifier using CNN features is competitive with retraining the deep convolution network [Karayev et al., 2013; Razavian et al., 2014; Yosinski et al., 2014]. This learning scheme is efficient enough to handle web-scale training data and thousands of target classes.

We first learn *synset* classifiers to estimate the probability that a synset s appears in an image with features \mathbf{x} :

$$p(s|\mathbf{x}) = \sigma(\mathbf{w}_s^T \mathbf{x}) \quad (3.3)$$

Here σ is the logistic function $\sigma(x) = 1/(1 + e^{-x})$, and \mathbf{w}_s is a weight vector trained to distinguish synset s from all other synsets. We learn one classifier per synset.

3.4.1.2 Defining a Concept’s Name Vocabulary

Before we can select names we must define a name vocabulary \mathcal{T}_s for each concept. For each synset s , we define a set \mathcal{T}_s that contains all words that can be used as the *name* for the semantic concept s . For example, \mathcal{T}_{cow} would include: *cattle*, *bovine*, and *animal* which are the more general categories; *cow* the basic-level name; and *kine* an archaic plural. This makes naming an image x into a multi-class classification problem. We estimate the probability of each choice $p(y_i^s|\mathbf{x}, s, u_s)$, for $i = 1, \dots, |\mathcal{T}_s|$, such that $\sum_i p(y_i^s|\mathbf{x}, s, u_s) = 1$.

We construct the name vocabulary \mathcal{T}_s by *tracing* the WordNet hierarchy up to 5 hypernym (parent concept) levels and extracting lemmas at each level (e.g., *riding horse* and *mount* are both lemmas of the same synset). The final set \mathcal{T}_s is the union of these lemmas. Due to its construction using the WordNet hierarchy, we will refer to set \mathcal{T}_s as the *trace* of synset s . By excluding the hyponyms (i.e. children nodes) of a synset from the *trace*, we differentiate context-dependent naming from fine-grained classification (e.g., distinguishing a *male horse* from a *mare*). Excluding names from sibling or other related nodes outside the direct inheritance line reduces the number of irrelevant names (e.g. *zebra* and *mule* are not acceptable names for *horse*).

We found that defining the name vocabulary using the WordNet hierarchy in this manner gave fewer irrelevant names than selecting the most frequent words in concept-caption pairs. However, this comes with a recall trade-off, one that may be mitigated by including names close to the synset of interest, with respect to word embedding or WordNet distance – we leave this for future work.

3.4.1.3 Naming Visual Concepts

There are two steps to naming once we know the visual concept. The first is deciding if the concept will be named. Many visual concepts, particularly those that define the setting, are not named in images because they are not considered important. Other authors have examined this problem specifically [Spain and Perona, 2011; Berg et al., 2012], and found that context can be used to predict importance. The second step is choosing a name for the concept, which we cast as a multinomial classification problem. Though we could have included separate class in this multinomial classifier to indicate that the concept was not described, we separate the two problems for convenience and efficiency. Identifying importance is performed on all images with a positive visual concept detection, while name selection need only be performed on the much smaller set of images where the visual concept is considered important.

Given a detected visual concept with synset s , we use a generative process to model the probability that the i 'th name $y_i^s \in \mathcal{T}_s$ is used to describe s , where \mathcal{T}_s is the name vocabulary described in Section 3.4.1.2. First, given the image and concept, we generate a switch variable u_s that indicates whether s is described at all. If u_s is *on*, we generate a name y_i^s from the distribution $p(y_i^s | \mathbf{x}, s = 1, u_s = 1)$. Here we assume each concept contributes support to only one name; in other words, a caption names each visual concept at most once. Overall, the probability of generating a name y_i^s given a concept and image features can be written as,

$$\begin{aligned} p(y_i^s | \mathbf{x}, s) &= \sum_{u_s \in \{0,1\}} p(y_i^s | \mathbf{x}, s, u_s) p(u_s | \mathbf{x}, s) \\ &= p(y_i^s | \mathbf{x}, s, u_s = 1) p(u_s = 1 | \mathbf{x}, s). \end{aligned} \quad (3.4)$$

Using this distribution we can trivially select the most likely name for each individual concept s . To get the most likely names for the whole image, we generate name-concept pairs based on these concept specific probabilities and globally rank them, as described in Section 3.4.1.4.

The following describes how we model $p(u_s = 1 | \mathbf{x}, s)$ and $p(y_i^s | \mathbf{x}, s, u_s = 1)$.

Is the concept described? In accordance with the generative model for concept names defined by Equation 3.4, we learn an *is_described* classifier to estimate the likelihood of synset s being explicitly described, given that it is visually present.

$$p(u_s = 1 | \mathbf{x}, s) = \sigma(\mathbf{w}_{u_s}^T \mathbf{x}) \quad (3.5)$$

With \mathbf{w}_{u_s} a learnable weight vector for each synset. Intuitively, the *is_described* classifier solves a similar problem to recent models for understanding object importance in

images [Berg et al., 2012]. We use CNN features \mathbf{x} , as they should capture most of the information related to concept importance, given their state-of-the-art performance in capturing scene types and contextual factors [Krizhevsky et al., 2012]. Moreover, CNN features can be extracted efficiently, which is necessary for scaling to large datasets.

How to describe the concept? The remaining part of Equation 3.4, the *description* classifier, is implemented as a one-vs-one linear SVC [Cortes and Vapnik, 1995] with multi-class probability estimates [Wu et al., 2004] based on platt scaling [Platt, 1999]. The probability of a synset s being described with name y_i^s rather than name y_j^s is (using the notation of Wu et al. [2004]):

$$p(y_i^s | y_i^s \text{ or } y_j^s, \mathbf{x}, s, u_s = 1) = \frac{1}{1 + \exp(A_{i,j}\bar{f} + B_{i,j})}, i \neq j \quad (3.6)$$

Where \bar{f} is the decision value from the SVM, while $A_{i,j}$ and $B_{i,j}$ are scalars learnt by cross-validation over the training set. Since we are using a one-vs-one scheme, we learn $A_{i,j}$ and $B_{i,j}$ for $\forall i, j : i < j$ with multi-class corrections as specified in Wu et al. [2004].

The three classifiers, *synset*, *is_described*, and *description*, are learned on successively smaller training sets increasingly tuned for finer grained descriptions, i.e. first identifying synset s , then $u_s = 0$ versus $u_s = 1$ only when s occurs, and finally choosing among \mathbf{y}^s when $u_s = 1$. In each case we use the same set of CNN features, but different training sets and different target classes. This improves training efficiency and allows us to train highly specialised classifiers.

For computational efficiency reasons, a concept s is considered for an image when $p(s|\mathbf{x})$ exceeds a high threshold. According to Equation 3.4, the *is_described* probability $p(u_s = 1|\mathbf{x}, s)$ is a scaling factor shared across all names y_i^s , so it does not affect name selection, but does apply when ranking names across synsets (Section 3.4.1.4).

3.4.1.4 Ranking Names and Concepts

Equation 3.4 describes the probability of generating a name for each synset s , but it does not stipulate how to rank names generated for different synsets. One way to impose a ranking is with the confidence of the *is_described* classifier $p(u_s = 1|\mathbf{x}, s = 1)$. However, we would like to take into account side information such as: description classifier reliability, concept occurrence prior likelihood, and the context imposed by other high-scoring name candidates.

We aim to learn a ranking score r for each triple composed of image features \mathbf{x}_i , synset s_m , word y_k , referred to by their respective indexes (i, m, k) . We use a linear

ranking function with weights \mathbf{w}_r :

$$r_{i,m,k} = \mathbf{w}_r^T h_{i,m,k}.$$

The optimisation problem for learning the ranking weights \mathbf{w}_r follows the RankSVM formulation [Joachims, 2002]. Training data is pairs of image-synset-word tuples (i, m, k) and (j, q, l) , where word k , synset m is associated with image i , while word l , synset q is *not* associated with image j , and $\xi_{i,m,k;j,q,l}$ are non-negative slack variables. C is a hyper-parameter representing the trade-off between training error and margin width.

$$\begin{aligned} \text{minimise : } J(\mathbf{w}_r, \xi) &= \frac{1}{2} \mathbf{w}_r^T \mathbf{w}_r + C \sum_{i,m,k;j,q,l} \xi_{i,m,k;j,q,l} \\ \text{s.t. } \forall (i, m, k; j, q, l) \\ \mathbf{w}_r^T h_{i,m,k} &\geq \mathbf{w}_r^T h_{j,q,l} + 1 - \xi_{i,m,k;j,q,l} \\ \xi_{i,m,k;j,q,l} &\geq 0 \end{aligned} \quad (3.7)$$

We use four types of features that capture information relevant to ranking image-synset-word tuples.

SCORES from different classifiers. These include: *is-described-score*, the probability that a synset is named given it is visually present Eq (3.5); *direct-to-noun-score*, the probability that a word k is used to describe image $\mathbf{x}_i - p(y_k|\mathbf{x}_i)$, obtained using logistic regression (this feature is also used as the *Direct-to-noun* baseline described below); *synset-score*, the probability that synset s_m is visually present in the image Eq (3.3).

Auxiliary information. This includes: *in-synset-frequency*, the prior of name k within the corresponding synset m ; *global-noun-freq*, the prior probability of word k in all training image captions; *description-accuracy*, the accuracy of the description classifier for this synset based on cross-validation performance; *is-described-accuracy*, the accuracy of the *is_described* classifier from cross-validation; *trace-size*, the number of words in the trace, previously denoted $|\mathcal{T}_s|$ in Eq (3.6).

KNN-RANK. We find image \mathbf{x}_i 's n -nearest neighbours in the training set and retrieve their captions. Nouns extracted from these captions are ranked with TF-IDF. By matching name k to a noun in the retrieved set, a *rank* for that name is established. We then define *knn-rank* as $1/\text{rank}$ or zero if the name failed to match any nouns. n is chosen to be 500 in this work.

WORD2VEC features are used to capture the word context. We use a modified version of the Word2Vec Continuous Bag of Words (CBOW) model [Mikolov et al., 2013]

without hierarchical softmax. Our CBOW training uses randomly selected context words from anywhere in the caption, rather than within a window of the target word. This applies to image caption training because the image semantically links all caption words: longer documents typically used to train word2vec do not have this property. Our CBOW model projects words into a 100 dimensional feature space.

We extract two different types of Word2Vec features from the set of candidate names for each image, broadly described as similarity and score features. *word2vec-similarity-max* and *word2vec-similarity-avg* are the maximum and average cosine similarity between word k and the most likely 6 words for image i according to *is_described* scores. *word2vec-score-max* and *word2vec-score-avg* are the maximum and average probability of the target word k given by the CBOW model when the context words are a random subset of high scoring name candidates for image i . The max and average are over 10 randomly sampled subsets. The probability estimates used here are a standard by-product of the CBOW model, which is trained by predicting a target word from its context.

We augment the ranking features by appending the products of all feature pairs. Augmenting the feature vector of linear-SVM is known to produce competitive performance compared to SVMs with non-linear kernels [Yuan et al., 2012]. We also added log transformed features but found that they did not improve performance. For each image i , we generate the final set of ranked names by removing duplicate names from the ranked list of synset-name pairs (m, k) , keeping the higher ranked pair.

3.4.2 Experimental Setup

3.4.2.1 Training and Testing Datasets

We use the IMAGENET-FLICKR [Xie and He, 2013] dataset to train synset classifiers for context-dependent naming in Section 3.4.3.1. This dataset is a subset of ImageNet[Deng et al., 2009], containing over 5.7 million images sourced from Flickr. Using the intersection between ImageNet and Flickr ensures all images have WordNet synset labels and Flickr metadata such as caption and tags.

We use the SBU dataset [Ordonez et al., 2011] to train and evaluate our *is_described* and *description* classifiers. This dataset consists of 1 Million Flickr images with associated captions; however, at the time of collection, only 95%, or 950K images were still publicly available. From the images obtained we reserve 2000 images for evaluation. These images are chosen because they form the two datasets, of 1000 images each, used by Ordonez et al. [2013], here referred to as SBU-1KA and SBU-1KB. SBU-1KA contains randomly selected images, while SBU-1KB contains images for which

Ordonez et al. [2013] had high confidence detections. We use 80% of the remaining images, or 760,000, for training the *is_described* and *description* classifiers; while 40,000, or 4% are used for training the rankSVM. The remaining 148,832 images are used for evaluation, which we refer to as SBU-148K.

Generating ground-truth descriptions for the SBU dataset We extract lemmatized nouns from the image captions and filter out nouns that are not part of ImageNet. The names in the resulting set are subject to label noise since some captions do not refer to visual content; however, they capture language use in a natural setting. In contrast, the naming exercise used to generate SBU-1KA and SBU-1KB enforces a less natural setting that explicitly strips away external image context and annotator intent.

3.4.2.2 Model Learning

The image feature \mathbf{x} is a 4096 dimensional vector from the second-to-last layer of a CNN [Jia et al., 2014] pre-trained on the ImageNet ILSVRC [Russakovsky et al., 2015] dataset. We learn 2633 new synset classifiers (Equation 3.3) on IMGNET-FLICKR; the chosen synsets have a large number of positive examples in the IMGNET-FLICKR dataset and are named in at least 100 SBU images. The threshold on the *synset* classifier is chosen to be $p(s|\mathbf{x}) > 0.95$.

The *is_described* classifier is trained on images representing the synset s according to the *synset* classifier. The positive images have captions containing a word in \mathcal{T}_s ; all other images in the subset are negative. Random sampling is used to select negative examples. For the final ranking of names we use SVM^{rank} [Joachims, 2006]. In all cases hyper-parameters are set with grid-search cross-validation.

3.4.2.3 Evaluation Metrics and Baselines

We calculate a precision-recall curve to evaluate our model by sorting the output names by confidence, before computing the precision and recall at each position in this list. The mean and standard deviation of these precision-recall points is calculated using 10 random testing set partitions. Precision and recall are the metrics reported by Ordonez et al. [2013], although they only provide precision and recall for the top 5 names per image rather than a full P-R curve.

The proposed image-to-name model is denoted *BasicName-Visual+Lang*. The proposed model sans the ranking approach of Section 3.4.1.4 is denoted *BasicName-Visual*; it uses *is_described* scores (Equation 3.5) for ranking. When performing per-synset evaluations, *BasicName-Visual* is equivalent to *BasicName-Visual+Lang* as the ranking only applies across synsets.

Four baselines are compared with *BasicName-Visual+Lang*.

- *Ngram-biased-SVM*, as presented by Ordonez et al. [2013]. This baseline only applies to the SBU-1KA and SBU-1KB datasets.
- *Direct-to-noun*, a method consisting of 2,549 separate logistic regressors predicting nouns y_i from CNN features \mathbf{x} . Because of class imbalance, we trained on a balanced dataset constructed by down-sampling negative examples, before calibrating the probabilities using held-out data.
- *Most-frequent name*, a method that outputs the most-frequent name y_i in each trace \mathcal{T}_s for synset s . Names are ranked across concepts by their frequency.
- *Frequency+described*, outputs the most-frequent name y_i in each trace \mathcal{T}_s for synset s . Names are then ranked across concepts using the *is_described* classifier (Eq. 3.5).

3.4.3 Results

We divide our evaluation into two parts. Section 3.4.3.1 evaluates name selection per synset, while Section 3.4.3.2 evaluates the complete image-to-name pipeline.

3.4.3.1 Name from Visual Context with a Known Concept

Multiple common names. We examine all 3,398 synsets with at least 200 captions from *ImageNet-Flickr* matching the trace $y_i \in \mathcal{T}_s$. Of these synsets 1,026, have a second common name: one used in at least 10% of image captions. Demonstrating semantic concepts with multiple names is a common phenomena.

Name from visual context. Using *BasicName-Visual* to detect synsets and choose names for images in the SBU dataset, we evaluate per-synset accuracy improvement over the *Frequency+described* baseline. The accuracy delta is shown in Figure 3.6, while Figure 3.7 gives the accuracy for each approach. Full accuracy results for each synset are published online³. Among the 2,633 frequent synsets in the SBU dataset, *BasicName-Visual* improves upon the *Frequency+described* baseline in 1,222 synsets, of which 783 improved by more than 1%. No change in accuracy is measured for 1190 synsets, while a small accuracy decrease affects 221 synsets.

BasicName-Visual provides the most improvement for synsets with ambiguous basic-level names: two or more names used with similar frequency. The fraction of improved synsets is on par with the fraction of synsets with multiple common names (see above). Similar to the pilot study in Section 3.3, we find the most improved synsets tend to be hard to name by frequency alone (see top of Figure 3.7), giving

³<https://github.com/computationalmedia/naming-with-visual-context>

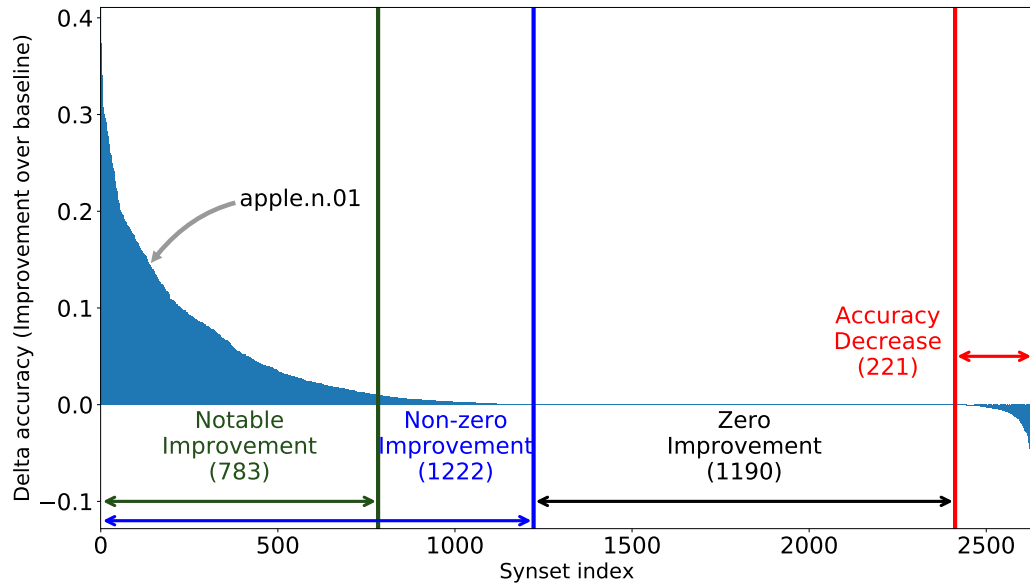


Figure 3.6: Per-synset accuracy improvement of *BasicName-Visual* over the *Frequency+described* baseline, ordered by accuracy delta. Compare to accuracy in Figure 3.7 (same x-axis order). See Section 3.4.3.1 for discussions.

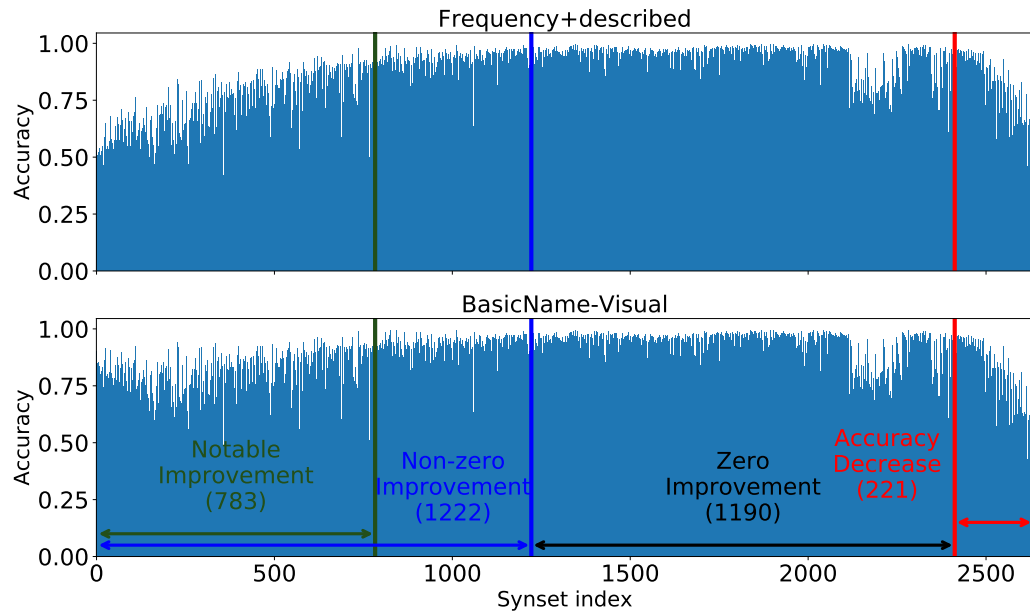


Figure 3.7: Per-synset accuracy for the *Frequency+described* baseline (top) and *BasicName-Visual* (bottom), ordered by accuracy delta. Compare to delta accuracy in Figure 3.6 (same x-axis order). See Section 3.4.3.1 for discussions.

	SBU-1K _A		SBU-1K _B	
	Precision	Recall	Precision	Recall
Ngram-biased+SVM	19.9 ± 1.2	10.4 ± 0.7	25.1 ± 2.4	14.4 ± 1.4
Direct-to-noun	20.0 ± 1.6	10.5 ± 0.9	21.5 ± 1.6	12.2 ± 0.7
Frequency+described	26.1 ± 1.7	12.8 ± 1.0	28.1 ± 1.3	15.2 ± 1.5
BasicName-Visual+Lang	26.3 ± 1.8	12.9 ± 1.1	28.9 ± 1.3	15.6 ± 1.2

Table 3.1: The precision and recall at 5, evaluated on the SBU-1K_A and the SBU-1K_B datasets. This shows *BasicName-Visual+Lang* outperforming the Ordonez et al. [2013] approach, *Ngram-biased+SVM*, in terms of both precision and recall.

rise to a mean accuracy of 0.65, improved to 0.73 with context. For comparison, the synsets where context gave zero accuracy improvement have a mean accuracy of 0.88. In these cases the basic-level name is likely a reliable approximation. The 221 synsets that exhibit an accuracy decrease are characterised by multiple names with similar frequency and fewer than average training examples. The similar name frequency ensures reliance on the visual context, but the visual classifiers are weak because of the limited training examples. Given more training examples, these 221 synsets could be improved, since their name distribution is conducive to contextual classification.

For the synsets which showed accuracy decrease with our method, a prior favouring the most frequent name could be beneficial. In a hierarchical probabilistic framework this prior could be learnt across all synsets, effectively deciding when the basic-level name is appropriate. We leave this as a direction for future work.

Illustrative examples. Figure 3.8 shows results of *BasicName-Visual* in comparison to labels from Mechanical Turk workers [Ordonez et al., 2013] and N-gram frequency [Ordonez et al., 2013]. Figure 3.8 shows synsets for which prior work [Ordonez et al., 2013] has not provided names. We see several aspects of visual context come into play when choosing names – including view point variation (*plant* vs *tree*; or *bird* vs *heron*); the presence of other object or part (*apple* vs *fruit*; *door* vs *screen*); and the appearance variations within the category (*art*, *sculpture* vs *carving*).

This is the first large-scale, fully automatic, classification of names via visual context. Our use of image collections with objects in their natural context enables us to discern context-dependent names previously observed in controlled lab environments [Barsalou, 1982; Mareschal and Tan, 2007], and alleviates the need for crowd-sourced labels [Ordonez et al., 2013].

3.4.3.2 Image to Names

We evaluate the performance on the image to names task with the three SBU test subsets described in Section 3.4.2. Precision-recall curves are generated using the $k \in 1, 2, \dots, |\mathcal{T}_s|$ highest ranked names per image.

Figure 3.10 compares *BasicName-Visual* to four baselines on SBU-1KA (left) and SBU-1KB (right). When predicting 5 words per image (i.e. $k = 5$) on SBU-1KA, *BasicName-Visual* achieves a precision of 0.26 at 0.13 recall, which is an improvement over *Ngram-biased-SVM* [Ordonez et al., 2013], with a precision of 0.20 at 0.10 recall – see Table 3.1. *BasicName-Visual* also achieves better performance than both *Most-frequent name* and *Frequency+described*. We find *Ngram-biased-SVM* is on par with *Direct-to-noun*. The same evaluations are carried out on SBU-148K (Figure 3.11). With two orders of magnitude more testing data – hence lower variance for performance estimates – we can measure a significant difference between *BasicName-Visual*, *Most-frequent name* and *Frequency+described*. A result demonstrating *is_described* and *description* both contribute to naming accuracy.

Figure 3.11 shows *BasicName-Visual+Lang*, which uses language context and auxiliary information for ranking names. This approach out-performs *BasicName-Visual* across the recall range. The average precision of *BasicName-Visual* is 0.336 ± 0.003 , improving to 0.341 ± 0.002 with SCORES features, and further improved to 0.347 ± 0.002 with KNN, WORD2VEC and AUX features. This result demonstrates each feature set is important for ranking names.

Figure 3.12 shows several examples of the image to name task. In the first four rows, *BasicName-Visual+Lang* correctly chooses a more specific name than the one chosen by frequency alone, such as preferring *church* over *building* in row 3 and *sunflower* over *flower* in row 4. Row 5 contains an example where synset classifiers break down. Here the main objects (bikes and people) are small and subject to poor lighting. Row 6 shows a difficult case where the scene contains several objects that do not usually co-occur. The prediction *ball* can be considered correct but is not in the ground-truth.

In the next section (Section 3.5) we continue our analysis of naming by identifying naming patterns in a hierarchy.

3.5 Large Scale Naming Patterns in a Taxonomy

In previous sections naming was cast as a flat decision problem: once names for each concept were identified, the hierarchical relationships were removed. This was demonstrated as a reasonable approximation; however, hierarchical relationships are

still valuable. They could, for example, allow naming specificity to fit an individual’s domain knowledge. Alternatively, under additional stylistic constraints such as sentiment, a hierarchy could help to trade off stylistic conformance with visual specificity. In general, hierarchical relationships offer a concise way of trading off specificity with another objective.

We explore the hierarchical structure of naming using a large image-caption dataset, and focus on understanding the specificity of animal naming. As a concrete example, we test and refine the theory presented by Lakoff [Lakoff, 1987], stating that animal naming at the level of genus is most common. This theory is based on the original definition of genus as the level where subcategories are visually distinctive. Genus sits just above species: defined by interbreeding possibilities. The species that survive in a geographic region are those best adapted, often leaving one species per genus per geographic region, making different genus locally distinctive.

Our analysis employs automatic visual detection of animals organised by the Linnaean hierarchy. The presented techniques are general enough to be employed in any object naming domain where specificity is defined in terms of depth in a hierarchy (e.g. in the animal *kingdom* the *genus* is at the same level in the hierarchy regardless of the *subspecies*) – true of many scientific domains. We further validate our automatic method with a crowd-sourced Amazon Mechanical Turk (AMT) naming experiment. Finally, using visualisations we explore the impact of, and potential biases introduced by, automatic concept detectors.

Results indicate that naming patterns can be identified on a large scale, but contrary to the conventional wisdom in cognitive psychology [Lakoff, 1987] they are not dominated by *genus* for animals. We observe that across a few hundred classes of mammals, reptiles and birds, the level of specificity for naming differs from concept to concept. We also note that the quality of concept detectors has a large influence on the inferred names.

3.5.1 Datasets and Pre-processing

Concept Hierarchy We use the Integrated Taxonomic Information System (ITIS) ⁴ to define the concepts in the animal kingdom. ITIS consists of a taxonomy for plants, animals, fungi and microbes around the world and is developed and supported by federal agencies in the United States. When this research was conducted, the system had over 690000 scientific names and 124000 common names arranged hierarchically by their classifications called taxa (singular, taxon) e.g. kingdom, class, genus and species. In ITIS the depth of a taxon has a consistent interpretation (eg class, genus or

⁴<http://www.itis.gov>

species) which relates to the specificity. This makes it possible to draw conclusions about the specificity of naming choices across different animal types. In contrast, WordNet does not define specificity in a consistent way across sub-trees.

Image Captions We use the SBU 1-Million image-caption dataset [Ordonez et al., 2011], which was sourced from Flickr and filtered for captions with linguistic features indicative of visual relevance.

Any sub-string of the caption is a candidate name for the visual concept; however, we only use uni-grams or bi-grams that match a node in the taxonomy. When overlapping n-grams match the taxonomy, the more specific name – as defined by depth – is chosen. For example, in the case of *eagle* and *bald eagle* we select the more specific bi-gram *bald eagle*. An n-gram is matched to a node in the taxonomy using exact string matching to vernacular names or scientific names; word concatenation, lemmatization and punctuation removal are used to improve recall.

3.5.2 Model

Given an image in the SBU dataset, we automatically detect the concept label with a pre-trained classifier. Specifically, we identify the ImageNet synset labels using the pre-trained Oxford VGG 16-layer network [Simonyan and Zisserman, 2015]. This network is trained to classify 1000 different visual synsets. We map these synsets to nodes in the animal taxonomy by matching strings from synset lemmas to taxonomy vernaculars and names. The resulting mapping is many-to-many, though typically taxonomy entries only have one synset mapped to them. We only consider the most confident, i.e., top one, visual prediction of the VGG network for each image. We filter out images with a top one visual concept that is not an animal.

Matching objects to names. We first select a sub-tree of the taxonomy such as *Mammalia* (Mammals), *Aves* (Birds) or *Reptilia* (Reptiles). For each image we match the highest confidence visual concept to names in the caption. We require that both the visual concept and the possible name map to taxonomy entries in the sub-tree of interest, and the name and the visual concept have a descendant or ancestor relationship. These conditions ensure the classifier and caption agree on the concept, providing confidence that a name actually refers to the visual concept.

3.5.3 Results

The results are divided into three subsections: large scale naming patterns (Section 3.5.3.1), human evaluations (Section 3.5.3.2), and concept detector performance (Section 3.5.3.3). In Section 3.5.3.1 we draw conclusions about naming specificity across different animal classes using the results of our large scale naming study on image-

caption pairs with automatic concept detectors. In Section 3.5.3.2 we conduct a small scale human evaluation task to validate our findings – our analysis touches on the importance of context and motivation in naming. Finally, In Section 3.5.3.3 we use a powerful visualisation approach to explore concept detector errors.

3.5.3.1 Large Scale Naming Patterns

Using automatically detected concepts and names matched to the ITIS taxonomy, we explore how different classes of animals such as birds, mammals, reptiles, are named. We count both the frequency of concept-name pairs and concept-taxon level pairs. Name frequency per-concept gives a fine grained view of concept descriptions. Normalising taxon counts for each concept (ie down columns in Figure 3.13) provides an overview of naming specificity in captions.

Using the SBU 1-Million image-caption dataset, we calculate the level at which each animal concept is named. There are over 59000 images in the SBU dataset with an animal name in the caption that matches the visual classification via a descendant or ancestor relationship. Figure 3.13 shows the results for the *Mammalia* class. For this subset of mammals the majority are frequently named at only one level of specificity, a result consistent with the theory of basic-level naming [Rosch et al., 1976]. Some mammals, however, are frequently named at multiple levels of specificity – a result that is incongruous with basic-level naming. For example *black bear* and *bear* are used with a similar frequency when naming *Ursus americanus*. This supports our findings from Sections 3.3 & 3.4, indicating that a single basic-level is not universally appropriate.

Figure 3.13 shows that animals in the *Mammalia* class are commonly described at the level of species, genus or family. *Aves* are typically described at the level of class by the name *bird* or occasional at the level of family or genus – Figure 3.14. *Reptilia* are typically described at the level of order or genus – Figure 3.15.

We observe animals in the class *Mammalia* are described more specifically than birds or reptiles. Mammals, unlike birds or reptiles, often have obvious shape differences, so our results are consistent with the idea that distinctive shapes are important in categorisation and naming [Lakoff, 1987]. Moreover, the most specifically named classes of birds and reptiles tend to be large with a distinctive shape such as: ostrich, black swan, alligator and iguana.

Interactive figures for *Mammalia* (mammals), *Aves* (birds) and *Reptilia* (reptiles) are provided online⁵

⁵<https://github.com/computationalmedia/naming-with-visual-context>

Animal	MTurk Names	SBU Names
<i>Dasyurus</i>		
<i>Bos</i>	ox	cattle
<i>Canis lupus</i>	wolf, dog	wolf, dog
<i>Ursus arctos horribilis</i>	bear	brown bear, bear
<i>Cebus capucinus</i>		capuchin monkey
<i>Marmota</i>	squirrel	
<i>Ailurus fulgens</i>	red panda	red panda
<i>Elephas maximus</i>	elephant	elephant
<i>Vulpes lagopus</i>	arctic fox, fox	
<i>Panthera leo</i>	lion	lion

Table 3.2: Common names selected by AMT workers for each animal. Names are in order from most frequent to least (left to right). The table shows names that occur in at least 10% of cases with a matching name, and with the total count greater than 20.

As a result of this filtering some cells are empty.

3.5.3.2 Human Evaluation

We conduct a small scale animal naming experiment on Amazon Mechanical Turk (AMT) to validate the large scale naming results in Section 3.5.3.1. We ask AMT workers to label 30 images for each of 10 animal categories with the name they would use to describe the animal. Three different AMT workers are assigned to each image, giving a total of 900 judgements. These judgements are then matched to the animal taxonomy and filtered as in Section 3.5.1.

The names chosen by turkers, shown in Table 3.2, demonstrate that some animals have multiple names in common usage (eg *arctic fox* and *fox*). Moreover, the results are similar to those obtained automatically from the SBU dataset. For example *Canis lupus*, *Panthera leo*, *Elephas maximus* and *Ailurus fulgens* have the same most common name in both the automatic and human evaluations. In the case of *Ursus arctos horribilis* (brown bear), turkers tended to use more general names than those used in the image-caption corpus. We hypothesise that photo up-loaders use more specific names because of their additional contextual information.

The empty cells in Table 3.2 represent cases where no name was chosen more than 10% of the time, or where names above this threshold did not match to ITIS. For example, *Cebus capucinus* (white-headed capuchin monkey) and *Dasyurus*. *Dasyurus* is a nocturnal marsupial native to Australia and New Guinea, so it is reasonable to assume annotators failed to identify it correctly. *Cebus capucinus* was overwhelmingly described as a *monkey*, though according to ITIS it is a *new world monkey*. This disconnect between the ITIS vernaculars and the names being used by turkers is the

reason *Cebus capucinus* did not have a name match.

Our human evaluations support the results of the large-scale automatic evaluation, with many names selected by annotators also identified by the automatic method. The evaluation also highlights the differences between controlled naming experiments and studying naming ‘in the wild’, where annotators have additional context and latent motivations. In a Mechanical Turk setting the homogeneous context and motivation triggers different naming choices.

3.5.3.3 Concept Detector Visualisations

Our concept detector is a state-of-the-art CNN (circa mid 2015), with a top-1 performance of approximately 70% [Simonyan and Zisserman, 2015]. To qualitatively observe the relationship between names and visual appearance, we build per-concept visualisations relating visual features and names. Specifically, we employ a t-Distributed Stochastic Neighbour Embedding (t-SNE) [der Maaten and Hinton, 2008], an unsupervised approach for embedding feature vectors into a low dimensional space, commonly used for visualising high-dimensional data. Our original feature space is 4096 dimensional and extracted from the second last layer of the VGG 16-layer CNN, while our transformed feature space is the 2-dimensional x,y-plane.

The t-SNE embedding for *Ursus maritimus* (polar bear) shown in Figure 3.16 divides the images into at least three distinct regions. In the upper right are polar bears in icy environments, in the lower left are polar bears swimming in the water, and in the middle are polar bears in enclosures or other environments. The name *polar bear* is used relatively uniformly throughout the space, while the name *bear* is primarily used in the middle section and generally not in the upper right hand corner where the polar bears are in icy environments. This indicates that people are less likely to name *Ursus maritimus* as *bears* when they are shown in a visually icy context.

The t-SNE embedding for *Cygnus atratus* (black swan), Figure 3.17, shows a number of classifier failures. All the images in this figure were classified as *Cygnus atratus*. The upper right of the figure shows white swans, the lower right shows ducks, while the left of the figure is mostly black swans. It is clear from this that the *Cygnus atratus* is typically described as a *black swan* and that the other names *duck* and *swan* are mostly spurious detections. The names *duck* and *swan* slipped past the caption matching procedure because in ITIS they are different possible names for *black swan*.

The presented t-SNE embeddings come from the CNN features also used for classification, so the separation between correct classifications and errors is interesting. This suggests that, although the classifier was not trained to differentiate between these closely related animal classes, the learnt features are nevertheless discrimina-

tive. We could eliminate some classifier errors by training new classifiers with the existing features.

3.6 Summary

This chapter focused on selecting visual concept names that are appropriate for image captions, and analysing naming choices in image captions. These focal points relate to two key challenges in stylistic image captioning: representing style and content, and overcoming data scarcity.

I develop a concept naming method that takes into account visual context and works at a large scale, without manual labelling. This method uses name candidates selected from WordNet and two separate classifiers for each concept: the name classifier and the is-described classifier. Using this solution to the concept-to-name task I tackle the image-to-name task with a three stage pipeline: automatic concept detection, concept naming, and name ranking. These approaches for concept-to-name and image-to-name show good performance in their respective subtasks. The result is the first catalogue of contextual names for thousands of visual concepts generated automatically using hundreds of thousands of images.

Basic-level names are insufficient to capture the complexities of naming and categorisation. Many visual concepts have more than one name in common usage: 60% of MSCOCO concepts and 30.2% of ImageNet concepts on SBU. Visual context helps to choose names, improving naming accuracy by more than 5% for 11.3% of MSCOCO concepts and for 15.7% of SBU concepts. Unlike smaller scale experiments in the cognitive psychology literature, I show that context is important in a natural setting for thousands of concepts. Aspects of visual context identified as important are: view point, concept co-occurrences, and category appearance variation. Patterns of naming within a hierarchy were also investigated on a large scale via an entirely automatic method. Results indicate different levels of specificity used to name mammals, reptiles, and birds – a result consistent with ideas from cognitive psychology. Contrary to the conventional wisdom in cognitive psychology [Lakoff, 1987], the animal names are not dominated by *genus*, although caution is advised when interpreting these results due to the classification inaccuracies identified.

This chapter relates to two key challenges in stylistic image captioning: representing style and content, and overcoming data scarcity. Style was previously defined as *how* text is written rather than *what* is communicated, and expressed through a set of consistent and distinguishable linguistic choices. In this chapter I controlled for the *what*, the visual concepts, and modelled the *how*, the naming choice. I showed that this concept level representation resulted in more natural, contextually appro-

priate names for concepts. However, only 30% of concepts have at least two common names, so introducing easily identifiable style into image captions seems to require structural changes beyond naming. A key advantage of the naming method presented here is its ability to learn on image-caption pairs mined from the web and thus to avoid expensive data annotation. Learning sub-components of a styled image caption generation system on data mined directly from the web has the potential to substantially reduce the burden of data collection.

Since the publication of the works on concept naming, there has been a shift towards end-to-end trainable models for image captioning. While in previous models the naming work fitted easily into captioning pipelines between object detection and surface realisation, end-to-end models cannot be so easily adapted. However, some recent state-of-the-art models [Fang et al., 2015; Wu et al., 2015; Gan et al., 2017b] do use separate vision components to extract semantic labels, rather than vectors of features. A visual concept naming pipeline such as the one developed here could fit more easily within this type of model, bringing advantages such as straightforward incorporation of domain knowledge and the ability to exploit incomplete or noisy training data. Similarly, visual concept naming pipelines could be applied to out-of-domain image captioning [Tran et al., 2016; Anderson et al., 2017; Anne Hendricks et al., 2016], where test images contain objects not seen during training.

Other interesting directions for future work include generalised trace construction by expanding candidate names beyond direct ancestors, and hierarchical models for sharing parameters across concept to name classifiers.





































Synset	mTurk	Ngram	Description Classifier				
california_poppy .n.01	flower	flower	flower				
			poppy				
screen_door .n.01	door	screen	door				
			screen				
boatbill .n.01	bird	bird	bird				
			heron				
white_ash .n.01	leaf	ash	plant				
			tree				
minivan .n.01	van	van	car				
			van				

Figure 3.8: Examples of context-dependent naming. For each synset we display crowd-sourced one-name-per-synset [Ordonez et al., 2013], n-gram based most frequent name [Ordonez et al., 2013], context-dependent names from *BasicName-Visual*, and four image examples for each name. For synsets without previous naming results see Figure 3.9.




































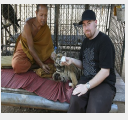




Synset	Description Classifier				
cathedral .n.02	building				
	cathedral				
	church				
apple .n.01	apple				
	fruit				
woodcarving .n.01	art				
	sculpture				
	carving				
tiger_cub .n.01	cat				
	tiger				

Figure 3.9: Examples of context-dependent naming. For each synset we display context-dependent names from *BasicName-Visual* and four image examples for each name. Unlike Figure 3.8, the synsets in this figure had no previous naming results available [Ordonez et al., 2013].

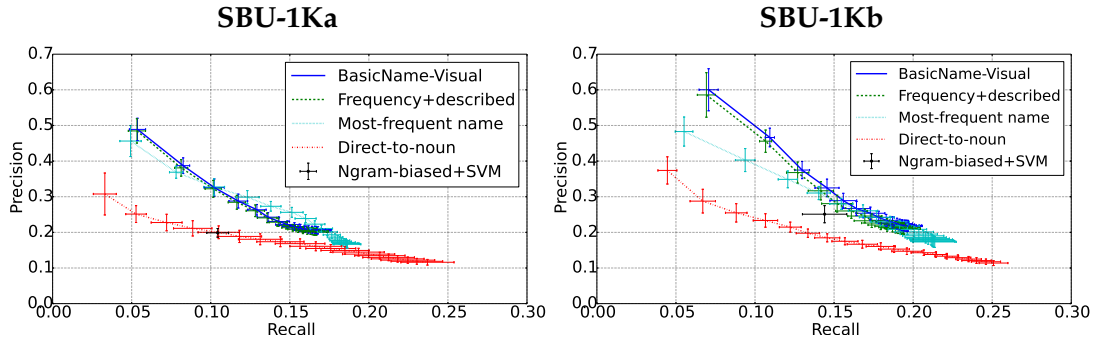


Figure 3.10: Precision-recall curves for our method and the four baselines on SBU-1Ka and SBU-1Kb. Error bars show one standard deviation.

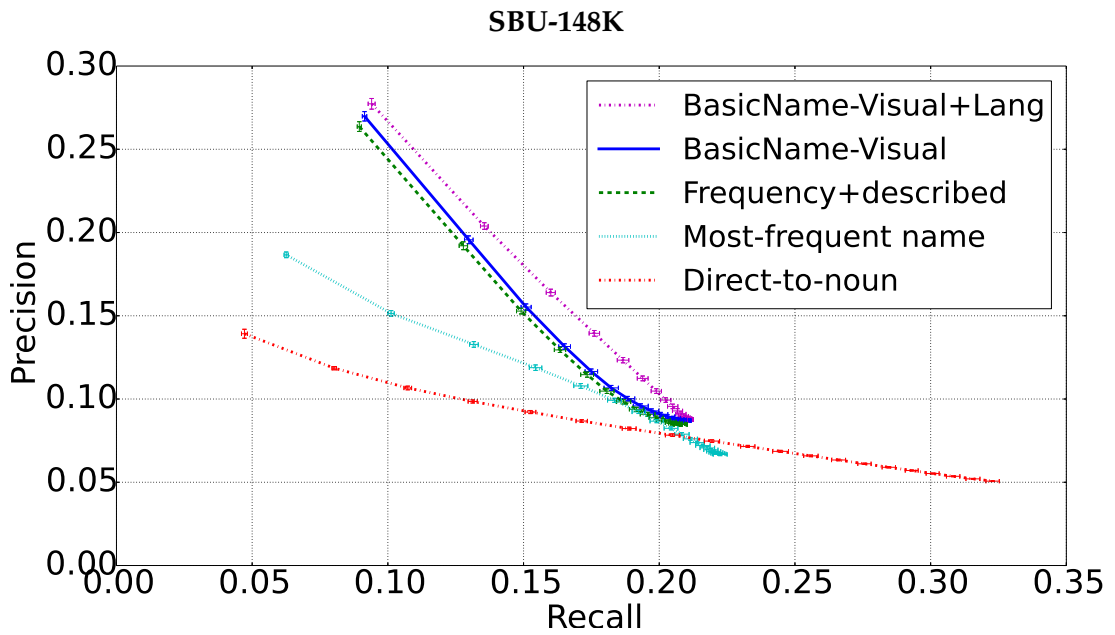


Figure 3.11: Precision-recall curves on SBU-148K. Error bars show one standard deviation.



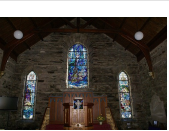

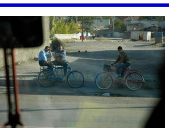

	Images	Labels	Ngram-biased-SVM	Direct-to-noun	Frequency+described	BasicName-Visual+Lang
Successful Examples		dirt, flower, grass, leaf, petal, plant, pot, rain, rise, rose, stem, white	flower tree plant dog face	tree window flower river house	plant rose white flower tree	rose white plant tree flower
		building, bush, field, tree, fountain, sky, grass, home, house, manor, white house, window, yard	building house home structure tree	tree house flower glass road	house grass building tree plant	house grass tree building road
		altar, alter, roof, art, building, flower, church, door, lamp, light, wall, podium, window, stain glass, vase, stain glass window,	street building door tower room	tree window house stained glass glass	window building tower monument column	window tower building church monument
		close-up, flower, petal, sky, tree, stamen, sunflower	bird pink tree plant white	house girl sign dog mountain	flower plant yellow	sunflower flower yellow plant
Failed Examples		adult, house, bicycle, bike, bush, parking lot, people, pole, street, tree, wall, window	neighborhood mountain zoo rock ground	flower bridge girl house car	car chair rider plant place	car seat person chair plant
		bubble, float, lake, man, pants, plastic, pond, shirt, shrub, water	shift dog zoo grass ball	water girl river beach house	ball fish building sail bird	ball fish building way sail

Figure 3.12: Example images from the SBU-1K_A and SBU-1K_B datasets with Amazon Mechanical Turk labels. We show the top names, predicted by our method, *BasicName-Visual*, and three baselines. Words printed in green match the hand labelled ground-truth. Our method performs well on the first four images but fails on the last two.

[illegible]

Figure 3.13: The specificity of names describing mammals. The first row is the most general taxon, while the last is the most specific. Each column is a different animal corresponding to a visual classifier. Darker colours indicate larger counts, with columns normalised. Columns with less than 20 detections were filtered out.

Figure 3.14: The specificity of names describing birds. Darker colours indicate larger counts, with columns normalised.

Subspecies	Species	Genus	Family	Order	Class	
			duck	water fowl	bird	Anatidae
	mergus serrator		duck	water fowl	bird	Mergus serrator
			heron		bird	Egretta caerulea
			hummingbird		bird	Trochilidae
			flamingo		bird	Phoenicopteridae
	strix nebulosa		owl		bird	Strix nebulosa
			parrot		bird	Psittacidae
			hawk		bird	Accipitridae
			heron		bird	Ardeidae
	grey parrot		parrot		bird	Psittacidae
			chickadee		bird	Paridae - erithacus
			hornbill		bird	Bucerotidae
	white stork		stork	cicones	bird	Ciconia ciconia
			stork		bird	Corvidae
			jay		bird	Phasianidae
			quail		bird	Bonasa
			partridge		bird	Perdix
			turkey		bird	Lyrurus
	black grouse		turkey		bird	Orididae
			bustard		bird	Fringillidae
					bird	Diomedelidae
					bird	Apodiformes
					bird	Gruiformes
					bird	Gruidae
					bird	Pycnonotidae
					bird	Pelecaniformes
					bird	Vidua
					bird	Circulidae
					bird	Aramus
					bird	Aramidae
					bird	Phoenicopteriformes
		swamp hen			bird	Galbulidae
		bulbul			bird	Porphyrio
		albatross			bird	Dionotus porphyrio
				water fowl	bird	Anseriformes
	limpkin				bird	Anseriformes
	indigo bunting		cardinal		bird	Passerina guarauna
	ruffed grouse		partridge		bird	Bonasa cyanea
		ptarmigan	quail		bird	Lagopus
	brambling				bird	Fringilla montifringilla
		chaffinch			bird	Junco
		junco	bunting		bird	Cinclus
		dipper			bird	Limnodynastidae
		prairie chicken	turkey		bird	Threskiornithidae
		magpie	crow		bird	Carduelis
		redshank		shorebird	bird	Carduelis
	tringa totanus			shorebird	bird	Carduelis
		oyster catcher		shorebird	bird	Carduelis
	ruddy turnstone	turnstone	sandpiper	shorebird	bird	Carduelis
			pelecanidae	heron	bird	Carduelis
			sandpiper	shore bird	bird	Carduelis
			sandpiper	shore bird	bird	Carduelis
				emu	bird	Carduelis
				emu	bird	Carduelis
				emu	bird	Carduelis
				shorebird	bird	Carduelis
				shorebird	bird	Carduelis
			sand piper	shore bird	bird	Carduelis
			ibis	heron	bird	Carduelis
		goldfinch	finch		bird	Carduelis
		goldfinch	finch		bird	Carduelis
		cockatoo	parrot		bird	Carduelis
	american coot	coot	water hen		bird	Carduelis
		robin			bird	Carduelis
	american robin	robin			bird	Carduelis
		pelican	pelecanidae	heron	bird	Carduelis
		crane			bird	Carduelis
			finch		bird	Carduelis
	house finch				bird	Carduelis
	bald eagle	fish eagle	eagle		bird	Carduelis
	king penguin		penguin		bird	Carduelis
ostrich				emu	bird	Carduelis
	black swan	swan	duck	water fowl	bird	Carduelis
		monarch			bird	Carduelis

Class	Order	Suborder	Family	Genus	Species
	turtle		sea_turtle		loggerhead
Caretta_caretta					
Anolis_carolinensis	lizard			anoie	green_anoie
Anolis	lizard			anoie	
Agama	lizard			agama	
Kinosternon	reptile			mud_turtle	
Emydidae	reptile		terrapin		
Kinosternidae	reptile				
Testudines	turtle				
Dermochelyidae	turtle				
Dermochelys_coracea	turtle				leatherback
Terrapene	turtle		terrapin	box_turtle	
Iguana_iguana	lizard			iguana	green_iguana
Iguana	lizard			iguana	
Thamnophis	lizard	snake		garter_snake	
Nerodia	lizard	snake		water_snake	
Alligator_mississippiensis	reptile				alligator

Figure 3.15: The specificity of names describing reptiles. Darker colours indicate larger counts, with columns normalised.

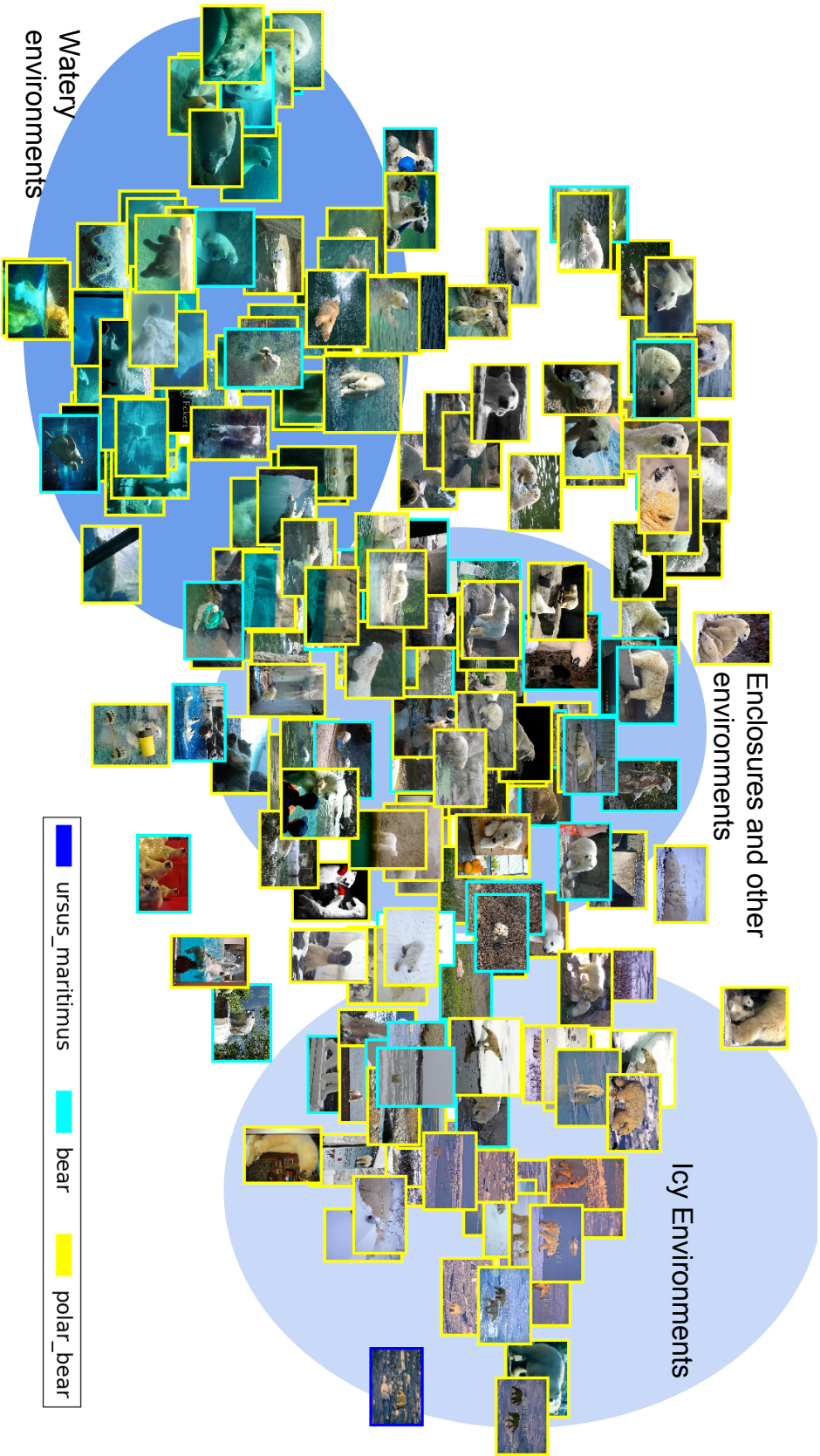


Figure 3.16: Visualisation of *Ursus maritimus* (polar bear) images using t-SNE with CNN features. Image border colours represent ground-truth names extracted from captions.

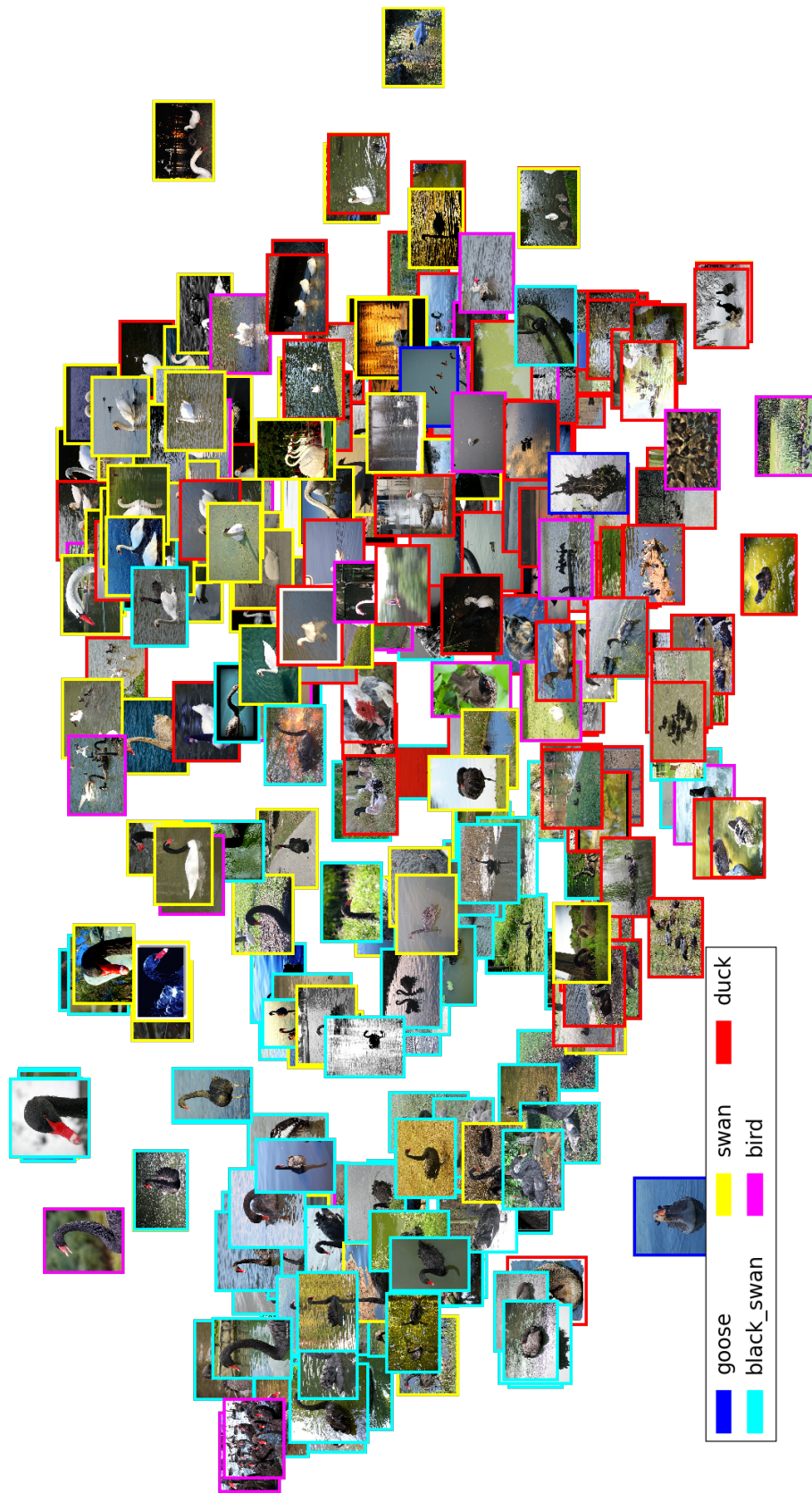


Figure 3.17: Visualisation of *Cygnus atratus* (black swan) images using t-SNE with CNN features. Image border colours represent ground-truth names extracted from captions.

Generating Image Captions with Strong Sentiment

4.1 Introduction

In this chapter I present SentiCap, an end-to-end stylistic caption generation system. Techniques in this chapter directly tackle the key task of modelling styled captions. My approach balances two models, one aimed at generating style, the other at visual description. I also propose transfer learning to reduce the effect of data scarcity in stylistic captioning, by learning from a large set of visually relevant captions and a small set of stylistic captions. I show how to crowd source stylistic captions, and elucidate the challenges faced. As part of the crowd sourced experiments I directly explore the freedom to choose a style for image captions. This tests the hypothesis that style is frequently an attribute of the image-caption pair, rather than being inherent to the image alone.

SentiCap focuses on the single stylistic attribute of sentiment, either positive or negative, as it simplifies data collection and evaluation. Sentiment is easily expressed in short captions, but can elicit a powerful emotional response from the reader. It is this response that makes generating captions with sentiment an important goal in its own right. The expression of clear and polarised emotions influences decision-making [Lerner et al., 2015] – from the mundane (e.g., making a restaurant menu appealing) to major (e.g., choosing a political leader in elections). For this reason, sentiment in natural language, has been extensively studied in the literature; however, related works focus on identifying or interpreting rather than generating sentiment. The methods I present for generating image captions are not tied only to sentiment, they apply more generally to style generation.

In Section 4.2 I explore two related areas: sentiment, which forms the problem setting; and transfer learning, which provides inspiration for the presented method. My approach is described in Section 4.3, and provides specific ideas towards the

key challenges of data scarcity and incorporating both style expression and visual relevance. The dataset construction in Section 4.4 demonstrates how to source a dataset consisting of stylistic captions; in doing so I further address the data scarcity challenge. Human evaluations of this dataset test the hypothesis that sentiment is an attribute of the image-caption pair, and is not intrinsic to most images. Experimental settings and results are in Section 4.5. Finally, Section 4.6 summarises the chapter and its relation to the key challenges.

4.1.1 Overview of the SentiCap System

SentiCap builds upon the CNN+RNN (Convolution Neural Network + Recurrent Neural Network) recipe that has seen many recent successes in image captioning [Donahue et al., 2015; Karpathy and Fei-Fei, 2015; Mao et al., 2015; Vinyals et al., 2015b; Xu et al., 2015a]. However, SentiCap is the first end-to-end model capable of generating image captions with a distinct style. In essence SentiCap is a switching RNN model which injects sentiment. Two parallel RNNs generate each sentence; one is a factual language model, the other specialises in sentiment, both are conditioned on the image. A novel word-level regularizer emphasises sentiment words during training, helping to define the optimal combination of RNN streams.

We gathered a new dataset of several thousand captions with positive and negative sentiments by re-writing neutral captions (Section 4.4). Trained on 2000+ sentimental captions and 413K neutral captions, our switching RNN out-performs a range of heuristic and learned baselines, generating more emotional captions, with greater automatic and human evaluation scores. In particular, SentiCap has the highest fraction of successes at injecting sentiment into the caption: 88% positive (or 72% negative) captions are perceived by crowd workers as more positive (or negative) than the factual caption, with a similar descriptiveness rating.

4.2 Related Work

SentiCap build upon recent neural image caption generation techniques; see Section 2.3 for a review of these, and related techniques. The Convolutional Neural Network component is further detailed in Section 2.1, and Recurrent Neural Networks for language generation are reviewed in Section 2.2.

The concept of linguistic sentiment is a vital part of SentiCap. I review the relevant literature in Section 4.2.2.

4.2.1 Transfer Learning

We formally define transfer learning following the presentation of Pan and Yang [2009]. A training set consisting of input space \mathcal{X} and output space \mathcal{Y} is denoted $\mathcal{D} = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$. In the transfer learning setting we have two datasets, the source \mathcal{D}^s sampled from distribution p^s , and the target \mathcal{D}^t sampled from distribution p^t . The goal is to find a function with high predictive accuracy on data drawn from p^t using both datasets \mathcal{D}^s and \mathcal{D}^t . This is reliant on p^s and p^t having similar properties, though we allow $p^t(y|x) \neq p^s(y|x)$, which is sometimes cited as a key difference between transfer learning and domain adaptation [Pan and Yang, 2009]. However, there is disagreement in the literature: some authors [Schweikert et al., 2008] label the $p^t(y|x) \neq p^s(y|x)$ case as domain adaptation, or present domain adaptation algorithms that are agnostic to this property [Daumé III, 2007]. In our case of styled caption generation we are interested in the case of fully labelled source and target data where $p^t \neq p^s$ and $p^t(y|x) \neq p^s(y|x)$.

There are a number of transfer learning algorithms based around the idea of using source examples that help to estimate the target and removing those that hinder estimation. Dai et al. [2007] apply a boosting method that iteratively re-weights samples from both the target and source. The target examples are weighted as per standard AdaBoost, while source examples receive a lower weight if they are misclassified. Liao et al. [2005] present a method for transfer learning in an active learning setting, where an auxiliary variable measures the mismatch between each source example and the target distribution. Wu and Dietterich [2004] present an SVM formulation that selects support vectors from the source dataset while minimising total classifier complexity and maximising classification accuracy on both source and target datasets.

Another popular approach for transferring knowledge between tasks is feature learning. This can take the form of: learning feature relevance [Lee et al., 2007], learning a new joint feature space on the source and target data, or learning a new feature space on only the source data. Lee et al. [2007] argue that some features are innately more relevant for prediction across multiple related tasks. They use a generalised linear model $P(y|x) = g(w^T x)$ where feature weights w are associated with Gaussian priors controlled by meta-features. These meta-features summarise the importance of each feature across all the training tasks. Raina et al. [2007] learn high level features from the source data with sparse coding. They then express the target data as a sparse linear combination of these features. A standard classifier such as SVM is applied to this new representation.

For transfer learning, it is often useful to transfer or share parameters between

models built on the source and target datasets. Taken to the extreme, models trained on each dataset can be weighted and combined to form the new model [Schweikert et al., 2008; Gao et al., 2008]. Another approach is to define a parameter prior to share knowledge between the tasks [Bacchiani and Roark, 2003]. Bacchiani and Roark [2003] adapt n-gram language models by using the source text to define a Dirichlet prior which informs parameter estimation on the target text. Alternatively parameters can be coupled in a regularisation term [Schweikert et al., 2008], such as:

$$R(\Theta) = \frac{\lambda_\theta}{2} \|\Theta^s - \Theta^t\|^2 \quad (4.1)$$

With hyper-parameter λ_θ , parameters Θ^t of the model built of the target data, and parameters Θ^s of the model built on the source data. In computer vision parameter transfer has proven particularly effective for CNN architectures [Razavian et al., 2014; Yosinski et al., 2014]. A CNN model is first trained to convergence on the source dataset, then the learning rates are reduced and it is fine-tuned to the target dataset. Later layers (closer to the output layer) are generally more task dependent so receive higher learning rates during fine tuning – early layers (closer to the input layer) may even be fixed. Similar approaches have been used for sentiment classification of online product reviews [Glorot et al., 2011].

SentiCap uses a mixture of different ideas from transfer learning. We couple parameters between the two language model streams using a regularisation term. We weight words based on their importance to each language model; this is defined via word level supervision rather than the statistical techniques common to transfer learning. We also use pre-trained word vectors to handle unknown words.

4.2.2 Sentiment

In the literature it is common to talk about sentiment analysis [Pang, 2006], which can be understood broadly as identifying opinions, subjectivity or emotion in texts, be they visual, written or spoken. Most sentiment analysis applications have a far more limited scope, for example determining if a written text portrays a product positively or negatively. In this chapter we focus on positive and negative sentiment in the multi-modal image-caption domain; to this end we review research on sentiment in both language and vision.

Identifying sentiment in text is an active area of research [Pang and Lee, 2008; Socher et al., 2013]. Several teams [Nakagawa et al., 2010; Täckström and McDonald, 2011] have designed sentence models with latent variables representing the sentiment.

Sentiment lexicon construction – identifying the average sentiment contribution of linguistic entities such as phrases or words – is a common sub-problem of sentiment classification. Popular approaches include: direct annotation at the phrase level [Esuli and Sebastiani, 2006; Taboada et al., 2011; Thelwall et al., 2010], inferring from document level annotations [Thelwall et al., 2010; Salvetti et al., 2006] and extrapolating from phrase relationships’ substructure [Esuli and Sebastiani, 2006; Hatzivassiloglou and McKeown, 1997]. SentiWordNet [Esuli and Sebastiani, 2006] is an attempt to provide positive, negative and neutral judgements to all WordNet synsets. Semi-supervised learning is used: the sentiment scores of un-seen words are approximated using WordNet relations to words with known sentiment. SentiWordNet is commonly used in the literature to support sentiment classification due to easy access, updates [Baccianella et al., 2010] and broad coverage. SentiStrength [Thelwall et al., 2010] is a lexicon based sentiment classification technique for short informal texts; it was built using MySpace comments, so the resulting lexicon contains common Internet slang and emoticons. The sentiment polarities were first specified by annotators at a word level, before being updated by a statistical method exploiting comment level sentiments.

The sentiment of an image is affected by context, so it helps to define different layers of sentiment. First person sentiment corresponds to emotions elicited by the author – often recorded for personal organisation reasons [Ames and Naaman, 2007]. Second person sentiment is expressed by individuals whom the photo is *communicated to*, often promoted by contextual information added by the author, or through shared experiences. Third person sentiment is expressed by an objective viewer, who lacks additional context. In the case of SentiCap we will be using third person sentiment: our annotators lack personal connections with the images and were not their original audience.

Researchers have explored how image presentation and content affects the viewers. Studies of image presentation and aesthetics [Murray et al., 2012; Joshi et al., 2011] have shown a correlation between photographic technique (e.g. high dynamic range, low depth of field) and emotional response. Machine learning models can detect photographic technique and predict the emotional response. However, only focusing on presentation and aesthetics misses a number of triggers, including image content and context. The Visual SentiBank [Borth et al., 2013; Chen et al., 2014] system focuses more on image content by employing a large Adjective-Noun-Pair (ANPs) detector catalogue. The chosen ANPs are frequent in online image captions and correlate with emotional response. This idea was further extended into the multilingual case [Jou et al., 2015], providing fine-grained emotional response as a function of culture. Although these systems are predictive rather than generative,

the idea of a visual sentiment vocabulary is essential for SentiCap. Our sentiment vocabulary used to guide annotation is built upon Visual SentiBank.

4.3 Generating Image Captions with Sentiment

The following sections describe SentiCap, a system for generating image captions with sentiment. Section 4.3.1 lays out the overarching model incorporating both style expression and visual relevance. The specific forms of individual components are discussed in Section 4.3.2.

4.3.1 Model Overview

Given an image I and its D_x -dimensional visual feature $\mathbf{x} \in \mathbb{R}^{D_x}$, our goal is to generate a sequence of words (i.e. a caption) $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ to describe the image with a specific style, such as expressing sentiment. Here $\mathbf{y}_t \in \{0, 1\}^V$ is 1-of- V encoded indicator vector for the t^{th} word; V is the size of the vocabulary; and T is the length of the caption.

We assume sentence generation involves two underlying mechanisms, one of which focuses on the factual description of the image while the other describes the image content with sentiment. We formulate this caption generation process using a switching multi-modal language model, which sequentially generates words. Intuitively, each generated word is either factual or sentimental, and the model must learn when to generate either word class; in doing so we are explicitly learning a trade off between semantics and sentiment. Formally, we introduce a binary sentiment variable $s_t \in \{0, 1\}$ for every word \mathbf{y}_t to indicate which mechanism is used. At each time step t , our model produces the probability of \mathbf{y}_t and the current sentiment variable s_t given the image feature \mathbf{x} and the previous words $\mathbf{y}_{1:t-1}$, denoted by $p(\mathbf{y}_t, s_t | \mathbf{x}, \mathbf{y}_{1:t-1})$. We generate the word probability by marginalising out the sentiment variable s_t :

$$p(\mathbf{y}_t | \mathbf{x}, \mathbf{y}_{1:t-1}) = \sum_{s_t} p(\mathbf{y}_t | s_t, \mathbf{x}, \mathbf{y}_{1:t-1}) p(s_t | \mathbf{x}, \mathbf{y}_{1:t-1}) \quad (4.2)$$

Here $p(\mathbf{y}_t | s_t, \mathbf{x}, \mathbf{y}_{1:t-1})$ is the caption model conditioned on the sentiment variable and $p(s_t | \mathbf{x}, \mathbf{y}_{1:t-1})$ is the probability of the word sentiment, with parameters \mathbf{W}^s .

We split the conditional language model $p(\mathbf{y}_t | s_t, \mathbf{x}, \mathbf{y}_{1:t-1})$ into two parts (illustrated in Figure 4.1), the factual model $p(\mathbf{y}_t | s_t = 0, \mathbf{x}, \mathbf{y}_{1:t-1})$, with parameters Θ^0 , and the sentiment model $p(\mathbf{y}_t | s_t = 1, \mathbf{x}, \mathbf{y}_{1:t-1})$, with parameters Θ^1 . This simplifies the underlying form and allows us to train each separately. We use a two-stage learning approach, first learning Θ^0 on a large dataset with factual captions and then learning

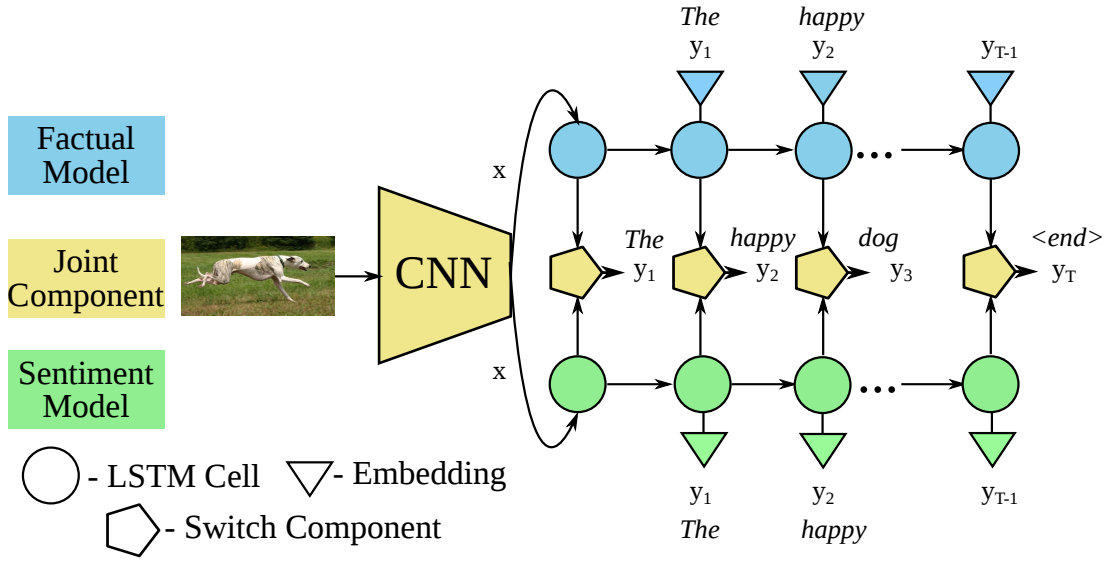


Figure 4.1: An overview of the SentiCap model. The factual language model is on top, the sentiment model is on the bottom. The switch component joins the two models, while the CNN is shared.

Θ^1 and \mathbf{W}^s jointly on a small set of sentiment captions.

Separated training of the factual and sentiment models is the key to dealing with limited training data. We transfer knowledge from the trained factual model to the sentiment model in the second training stage. This knowledge transfer is via a regularisation term which is discussed in Section 4.3.1.1.

4.3.1.1 Objective Functions

Our two stage learning approach requires an objective function for the factual conditional language model $p(\mathbf{y}_t | s_t = 0, \mathbf{x}, \mathbf{y}_{1:t-1})$, and a separate objective for the full switching model $p(\mathbf{y}_t | \mathbf{x}, \mathbf{y}_{1:t-1})$.

Objective for the factual conditional language model The factual conditional language models parameters Θ^0 are learned by minimising the negative log-likelihood of the caption words given images,

$$L^0(\Theta^0, \mathcal{D}^0) = - \sum_i \sum_t \log p(\mathbf{y}_{0,t}^i | s_t = 0, \mathbf{x}_0^i, \mathbf{y}_{0,1:t-1}^i) \quad (4.3)$$

The data is a large collection of factual image and caption pairs, denoted as $\mathcal{D}^0 = \{(\mathbf{x}_0^i, \mathbf{y}_0^i)\}_{i=1}^N$. This is a common loss function for training neural conditional language models – see Section 2.2.1.

Objective for the full conditional language model Utilising the trained factual language model in Eq (4.3), we jointly learn the parameters of the switching model \mathbf{W}^s and sentiment language model Θ^1 using a small image caption dataset with a specific sentiment polarity, denoted as $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i, \eta^i)\}_{i=1}^M$, $M \ll N$. Here $\eta_t^i \in [0, 1]$ is the sentiment strength of the t^{th} word in the i -th training sentence. As we train separate models for positive and negative sentiment η_t^i represents how strongly each word aligns with the target sentiment, with 0 being neutral and 1 indicating strong alignment with the target sentiment.

Our training objective incorporates word-level sentiment information to learning Θ^1 and the switching weights \mathbf{W}_s , while keeping the pre-learned Θ^0 fixed. For clarity, we denote the sentiment probability, parameterised by \mathbf{W}_s , as:

$$\gamma_t^0 = p(s_t = 0 | \mathbf{x}, \mathbf{y}_{1:t-1}) \quad (4.4)$$

$$\gamma_t^1 = 1 - \gamma_t^0 \quad (4.5)$$

The log likelihood of generating a new word \mathbf{y}_t given image and word histories $(\mathbf{x}, \mathbf{y}_{1:t-1})$ as $L_t(\Theta, \mathbf{x}, \mathbf{y})$, is formed by marginalising out the sentiment variable as in Eq (4.2). By also using Eq (4.5), we can rewrite this as:

$$\begin{aligned} L_t(\Theta, \mathbf{x}, \mathbf{y}) &= \log p(\mathbf{y}_t | \mathbf{x}, \mathbf{y}_{1:t-1}) = \\ &\log[\gamma_t^0 p(\mathbf{y}_t | s_t = 0, \mathbf{x}, \mathbf{y}_{-t}) + \gamma_t^1 p(\mathbf{y}_t | s_t = 1, \mathbf{x}, \mathbf{y}_{-t})]. \end{aligned} \quad (4.6)$$

The overall learning objective function for incorporating word sentiment, Eq4.8, is a combination of a weighted log likelihood and the cross-entropy between γ_t and η_t . The weighted log likelihood focuses on generating the correct word, while the cross-entropy ensures the word is generated from the most appropriate conditional language model, either factual or sentimental.

$$\mathcal{L}(\Theta, \mathcal{D}) = - \sum_i \sum_t (1 + \lambda_\eta \eta_t^i) [L_t(\Theta, \mathbf{x}^i, \mathbf{y}^i) \quad (4.7)$$

$$\begin{aligned} &+ \lambda_\gamma (\eta_t^i \log \gamma_t^{1,i} + (1 - \eta_t^i) \log \gamma_t^{0,i})] + R(\Theta), \\ R(\Theta) &= \frac{\lambda_\theta}{2} \|\Theta^1 - \Theta^0\|^2 \end{aligned} \quad (4.8)$$

where λ_η and λ_γ are hyper-parameters, and $R(\Theta)$ is a regularization term on the sentiment model with hyper-parameter λ_θ . Intuitively, when $\eta_t > 0$, i.e. the training sentence has a sentiment word at index t , the likelihood weighting factor $\lambda_\eta \eta_t^i$ increases the importance of L_t in the overall likelihood; at the same time, the cross-entropy term $\lambda_\gamma (\eta_t^i \log \gamma_t^{1,i} + (1 - \eta_t^i) \log \gamma_t^{0,i})$ encourages the switching variable γ_t^1 to

be > 0 , emphasising the sentiment language model. The λ_η term controls the importance of sentiment words relative to other words. λ_γ controls the importance of choosing the correct language model relative to minimising the negative log likelihood. This objective function captures the intuition that words have differing levels of importance for sentiment realisation; thus, mistakes on important words incur a larger loss than on less important words.

It is tempting to define the loss function as Equation 4.6 since it appears to trade-off between the two models and does not require the word level supervision of Equation 4.8. In practice, when a small sentiment training set is used, the sentiment word model $p(\mathbf{y}_t | s_t = 1, \mathbf{x}, \mathbf{y}_{-t})$ quickly overfits and the broader knowledge of the factual model becomes underutilised – $\gamma_t^{0,i} \approx 0 \forall i, t$. The cross-entropy term improves the generalisation error by introducing a penalty for using the sentiment word model for non-sentiment words. In place of cross-entropy, we considered using the squared difference $\frac{\lambda_\eta}{2}(\gamma_t^1 - \eta_t)^2$, but as explained in Section 2.1.1 the gradient tends to saturate, leading to slower learning.

The regularisation term $R(\Theta)$ (Eq (4.8)) trades-off between likelihood and parameter space differences between the sentiment and factual language models. This form of regularisation is a competitive approach to transfer learning [Schweikert et al., 2008]. Without this regularisation, we observe over-fitting from the sentiment language model even with the cross-entropy word level regularizer described previously.

4.3.2 Model Component Details

We adopt a joint CNN+RNN architecture [Vinyals et al., 2015b] in the conditional caption model. Our full model combines two CNN+RNNs running in parallel: one capturing the factual word generation, the other specialising in words with sentiment. The full model is a switching RNN, in which the variable s_t functions as a switching gate. This model design aims to learn sentiments well, despite data sparsity – using only a small dataset of image description with sentiments (Section 4.5.2). Hundreds of thousands of neutral image-sentence pairs [Chen et al., 2015] enable the learning of visual-text relationships.

Each RNN stream consists of a series of LSTM units. Formally, we denote the D -dimensional hidden state of an LSTM as $\mathbf{h}_t \in \mathbb{R}^D$, its memory cell as $\mathbf{c}_t \in \mathbb{R}^D$, the input, output, forget gates as $\mathbf{i}_t, \mathbf{o}_t, \mathbf{f}_t \in \mathbb{R}^D$, respectively. With k indicating the RNN

stream, the LSTM is defined as:

$$\begin{pmatrix} \mathbf{i}_t^k \\ \mathbf{f}_t^k \\ \mathbf{o}_t^k \\ \mathbf{g}_t^k \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T}^k \begin{pmatrix} \mathbf{E}^k \mathbf{y}_{t-1} \\ \mathbf{h}_{t-1}^k \end{pmatrix} \quad (4.9)$$

$$\mathbf{c}_t^k = \mathbf{f}_t^k \odot \mathbf{c}_{t-1}^k + \mathbf{i}_t^k \odot \mathbf{g}_t^k,$$

$$\mathbf{h}_t^k = \mathbf{o}_t^k \odot \mathbf{c}_t^k.$$

Here $\sigma(\chi)$ is the sigmoid function $1/(1 + e^{-\chi})$; \tanh is the hyperbolic tangent function; $\mathbf{T}^k \in \mathbb{R}^{4D \times 2D}$ is a set of learned weights; $\mathbf{g}_t^k \in \mathbb{R}^D$ is the input to the memory cell; $\mathbf{E}^k \in \mathbb{R}^{D \times V}$ is a learned embedding matrix in model k ; and $\mathbf{E}^k \mathbf{y}_t$ is the embedding vector of the word \mathbf{y}_t .

To incorporate image information, we use an image representation $\hat{\mathbf{x}} = \mathbf{W}_x \mathbf{x}$ as the word embedding $\mathbf{E} \mathbf{y}_0$ when $t = 1$, where \mathbf{x} is a high-dimensional image feature extracted from a convolutional neural network [Simonyan and Zisserman, 2015], and \mathbf{W}_x is a learned embedding matrix. Note that the LSTM hidden state \mathbf{h}_t^k summarizes $\mathbf{y}_{1:t-1}$ and \mathbf{x} . The conditional probability of the output caption words depends on the hidden state of the corresponding LSTM,

$$p(\mathbf{y}_t | s_t = k, \mathbf{x}, \mathbf{y}_{1:t-1}) \propto \exp(\mathbf{W}_y^k \mathbf{h}_t^k) \quad (4.10)$$

where $\mathbf{W}_y^k \in \mathbb{R}^{D \times V}$ is a set of learned output weights.

The sentiment switching model generates the probability of switching between the two RNN streams at each time t , with a single layer network taking the hidden states of both RNNs as input:

$$p(s_t = 1 | \mathbf{x}, \mathbf{y}_{1:t-1}) = \sigma(\mathbf{W}_s [\mathbf{h}_t^0, \mathbf{h}_t^1]) \quad (4.11)$$

where \mathbf{W}_s is the weight matrix for the hidden states.

An illustration of this sentiment switching model is in Figure 4.2. In summary, the parameter set for each RNN ($k = \{0, 1\}$) is $\Theta^k = \{\mathbf{T}^k, \mathbf{W}_y^k, \mathbf{E}^k, \mathbf{W}_x^k\}$, and that of the switching RNN is $\Theta = \Theta^0 \cup \Theta^1 \cup \mathbf{W}_s$.

Out of vocabulary words. The sentiment captions contain words absent from the factual captions. This presents two problems. First, it is unclear how to calculate the probability of the next word under the factual language model, $p(\mathbf{y}_t | s_t = 0, \mathbf{x}, \mathbf{y}_{1:t-1})$, in cases where $y \notin V$. Second, the regularisation term, Eq (4.8), requires the parameters of the two language models to have the same dimensions and underlying

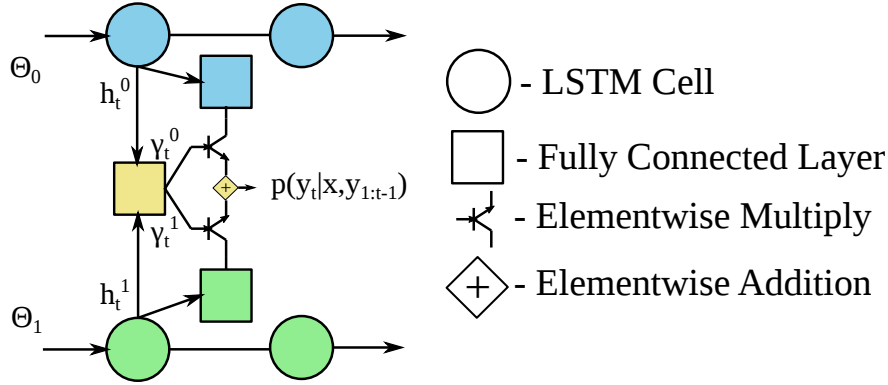


Figure 4.2: Illustration of the switching RNN model for captions with sentiment. LSTM cells are described in Eq 4.9. γ_t^0 and γ_t^1 are probabilities of sentiment switch defined in Eq (4.11) and act as gating functions for the two streams via the elementwise multiply blocks.

meaning.

Augmenting the factual language model vocabulary before training is not a practical solution since embeddings of words that were never seen during training do not get updated. The resulting embeddings would be random as per the initialisation strategy. Moreover, the fully connected output layer would learn that the new words are never generated and so minimise their output probability (with a cross-entropy objective this occurs by pushing up on the log probability of all other words).

We implement an alternative approach that uses pre-trained word vectors [Mikolov et al., 2013] to transfer knowledge from similar in-vocabulary words. Given a new word $y_n \notin V$, and pre-trained word embeddings E^{w2v} , we find the most similar word $\hat{y}_n \in V$ in the embedding space using Eq (4.12). We then extend the vocabulary and duplicating parameters in the learnt model corresponding to \hat{y}_n . Specifically, we set $E_{y_n}^0 := E_{\hat{y}_n}^0$ and $W_{y_n, \cdot}^{out,0} := W_{\hat{y}_n, \cdot}^{out,0}$, where E^0 are the embeddings learnt by the factual language model and $W^{out,0}$ are the learnt weights of the final fully connected output layer. Empirically this technique results in previously out of vocabulary words being generated by the switching model.

$$\hat{y}_n = \operatorname{argmin}_{y_m \in V} \frac{E_{y_n}^{w2v} \cdot E_{y_m}^{w2v}}{|E_{y_n}^{w2v}| |E_{y_m}^{w2v}|} \quad (4.12)$$

There are other possible ways to extend the vocabulary using pre-trained word vectors. I develop an alternative method in Section 5.3.2; however, this method is tuned for sentence simplification.

Settings for model learning. We use stochastic gradient descent with backpropagation on mini-batches to optimise the RNNs. We apply dropout to the input of

each step, which is either the image embedding $\hat{\mathbf{x}}$ for $t = 1$ or the word embedding $\mathbf{E}^k \mathbf{y}_{t-1}$ and the hidden output \mathbf{h}_{t-1}^k from time $t - 1$, for both the factual and sentiment streams $k = 0, 1$.

We learn models for positive and negative sentiments separately, due to the observation that either sentiment could be valid for the majority of images (Section 4.4.3). We initialise Θ^1 as Θ^0 and use the following gradient to minimise $\mathcal{L}(\Theta, \mathcal{D})$ with respect to Θ^1 and \mathbf{W}_s , holding Θ^0 fixed.

$$\frac{\partial \mathcal{L}}{\partial \Theta} = - \sum_i \sum_t (1 + \lambda_\eta \eta_t^i) \left[\frac{\partial L_t}{\partial \Theta} + \lambda_\gamma \left(\frac{\eta_t^i}{\gamma_t^{1,i}} \frac{\partial \gamma_t^{1,i}}{\partial \Theta} + \frac{1 - \eta_t^i}{\gamma_t^{0,i}} \frac{\partial \gamma_t^{0,i}}{\partial \Theta} \right) \right] + \frac{\partial R(\Theta)}{\partial \Theta}$$

Here $\frac{\partial L_t}{\partial \Theta}$, $\frac{\partial \gamma_t^{0,i}}{\partial \Theta}$, and $\frac{\partial \gamma_t^{1,i}}{\partial \Theta}$ are computed through differentiating across Equations (4.2)–(4.5). During training, we set $\eta_t = 1$ when word \mathbf{y}_t is part of an ANP with the target sentiment polarity, otherwise $\eta_t = 0$. We also include a default L2-norm regularization for neural network tuning $|\Theta|^2$ with a small weight (10^{-8}). We automatically search for the hyperparameters λ_θ , λ_η and λ_γ on a validation set using Whetlab [Snoek et al., 2012].

4.4 Constructing a Dataset of Captions with Sentiment

In order to learn the association between images and captions with sentiments, we build a novel dataset of image-caption pairs where the caption both describes an image and conveys the desired sentiment. In this section we summarise the new dataset and the crowd-sourcing task to collect image-sentiment caption data.

There are many ways a photo can evoke emotions. We focus on sentiments *from an objective viewer* who does not know the back story of the photo and is not trying to communicate to a particular individual – a setting also used by recent collections of objectively descriptive image captions [Chen et al., 2015; Hodosh et al., 2013]. Sentiments expressed by the author of the photo may rely on personal or shared context not contained in the photo itself – such contextual reasoning is out of scope.

We design a crowd-sourcing task to collect such objectively described emotional image captions. This is done in a caption re-writing task based upon objective captions from MSCOCO [Chen et al., 2015] by asking Amazon Mechanical Turk (AMT) workers to choose among ANPs of the desired sentiment, and to incorporate one or more of these into any one of the five existing captions.

4.4.1 Adjective Noun Pair Vocabulary Construction

Our ANP vocabulary comes from existing collections of ANPs associated with sentiment and from a large set of online image captions. For construction, we adopt a similar methodology to Visual SentiBank [Borth et al., 2013], a database of Adjective-Noun Pair (ANP) classifiers that are frequently applicable to online images. We take the title and the first sentence of the description from the YFCC100M dataset [Thomee et al., 2015], keep entries that are in English, tokenize these, and obtain all ANPs that appear in at least 100 images. We score these ANPs using the average of SentiWordNet [Esuli and Sebastiani, 2006] and SentiStrength [Thelwall et al., 2010], with the former being able to recognise common lexical variations and the latter designed to score short informal text. We keep ANPs that contain clear positive or negative sentiment, i.e., that have an absolute score of 0.1 and above. This gives us 1,027 ANPs with a positive emotion, 436 with negative emotions. This new ANP vocabulary has a greater overlap with the visual objects in the MSCOCO images than the original SentiBank vocabulary. We released this new ANP vocabulary online¹.

Our ANP vocabulary is slightly different to SentiBank [Borth et al., 2013]. SentiBank was constructed by searching Flickr (a photo sharing website) and youtube (a video sharing website) for adjectives from a pre-defined list. They then extract adjective noun pairs and sub-sample them to avoid dominance by a few popular adjectives. For the SentiCap vocabulary, we use a dataset of 100 million Flickr image captions from which we extract adjective noun pairs directly without sub-sampling. This means that the most common adjectives in our ANP vocabulary (adjectives that apply to the most nouns) tend to be less visually specific than SentiBank – see Table 4.1. Adjectives with low visual specificity are easy to introduce into captions and so are appropriate for SentiCap. Moreover, the SentiCap set of ANPs is closer to the underlying distribution of ANPs, which is a useful property for caption generation, whereas diversity is desirable for the SentiBank classification task. The most frequent nouns are similar between the two vocabularies, with “face”, “cat”, “dog”, and “baby” being frequent in both ANP sets. Overall, SentiCap uses 1463 ANPs with 212 unique adjectives and 322 unique nouns which is slightly more than SentiBank’s 1200 ANPs with 181 unique adjectives and 286 unique nouns.

4.4.2 Collecting Image Captions with Sentiment

We collect at least 3 positive and 3 negative captions per image. Figure 4.3 contains one example image and its respective positive and negative caption written by AMT workers. We released these captions online¹.

¹<http://cm.cecs.anu.edu.au/post/senticap/>

SentiBank		SentiCap	
<i>Adjective</i>	<i>Num. Nouns</i>	<i>Adjective</i>	<i>Num. Nouns</i>
colorful	20	great	119
beautiful	19	beautiful	101
tiny	18	nice	94
empty	18	good	93
abandoned	18	best	60
pretty	17	interesting	38

Table 4.1: The six adjectives in the SentiBank and SentiCap vocabularies that apply to the most nouns in the vocabulary. *Num. Nouns* is the number of nouns to which each adjective applies.



The painted train drives through a lovely city with country charm.

The abandoned trains sits alone in the gloomy countryside.

Figure 4.3: One example image with both **positive** and **negative** captions written by AMT workers.

We went through three design iterations for collecting relevant and succinct captions with the intended sentiment.

Our first attempt was to invite workers from Amazon Mechanical Turk (AMT) to compose captions with either a positive or negative sentiment for an image – which resulted in overly long, imaginative captions. A typical example is: “A crappy picture embodies the total cliché of the photographer ‘catching himself in the mirror,’ while it also includes a too-bright bathroom, with blazing white walls, dark, unattractive, wood cabinets, lurking beneath a boring sink, holding an amber-colored bowl, that seems completely pointless, below the mirror, with its awkward teenage-composition of a door, showing inside a framed mirror (cheesy, forced perspective,) and a goofy-looking man with a camera.”

We then asked turkers to place ANPs into an existing caption, which resulted in rigid or linguistically awkward captions. Typical examples include: “a bear that is inside of the great water” and “a bear inside the beautiful water”.

These results prompted us to design the following re-writing task: we take the available MSCOCO captions, perform tokenization and part-of-speech tagging, and identify nouns and their corresponding candidate ANPs. We provide ten candidate

ANPs with the same sentiment polarity and asked AMT worker to rewrite *any one of the original captions* about the picture using at least one of the ANPs. The form that the AMT workers are shown is presented in Figure 4.4. We obtained three positive and three negative descriptions for each image, authored by different Turkers. As anecdotal evidence, several turkers emailed to say that this task is “*very interesting*”.

The instructions given to workers are shown in Figure 4.4. We based these instructions on those used by Chen et al. [2015] to construct the MSCOCO dataset. They were modified for brevity and to provide instruction on generating a sentence using the provided ANPs. We found that these instructions were clear to the majority of workers.



 <p>Re-write one of the descriptions, using a word pair, to describe the image in a Positive way.</p> <p>Example Descriptions:</p> <ol style="list-style-type: none"> 1. a man swinging a bat during a baseball game 2. a baseball player bending over to hit a ball 3. a baseball player hitting a baseball at home base <p>Description <input type="text"/></p> <p>None of the word pairs are appropriate <input type="checkbox"/></p>	<ul style="list-style-type: none"> • Use the most appropriate of the word pairs below to describe the scene in a positive or negative way • Describe all the important parts of the scene. • Do not start the sentences with "There is". • Do not describe unimportant details. • Do not describe what a person might say. • Do not give people proper names. • The sentence should contain at least 8 words. <p style="text-align: center;">Word Pairs</p> <table border="1"> <tbody> <tr> <td>sunny field</td> <td>good man</td> </tr> <tr> <td>good game</td> <td>beautiful home</td> </tr> <tr> <td>great game</td> <td>clear field</td> </tr> <tr> <td>better home</td> <td>best man</td> </tr> <tr> <td>nice man</td> <td>great ball</td> </tr> </tbody> </table>	sunny field	good man	good game	beautiful home	great game	clear field	better home	best man	nice man	great ball
sunny field	good man										
good game	beautiful home										
great game	clear field										
better home	best man										
nice man	great ball										
 <p>Re-write one of the descriptions, using a word pair, to describe the image in a Negative way.</p> <p>Example Descriptions:</p> <ol style="list-style-type: none"> 1. a very small corner of a rest room with a toilet 2. a white toilet in front of a tiled bathroom wall 3. a bathroom with blue and white tiles and a white toilet <p>Description <input type="text"/></p> <p>None of the word pairs are appropriate <input type="checkbox"/></p>	<ul style="list-style-type: none"> • Use the most appropriate of the word pairs below to describe the scene in a positive or negative way • Describe all the important parts of the scene. • Do not start the sentences with "There is". • Do not describe unimportant details. • Do not describe what a person might say. • Do not give people proper names. • The sentence should contain at least 8 words. <p style="text-align: center;">Word Pairs</p> <table border="1"> <tbody> <tr> <td>cold water</td> <td>dirty wall</td> </tr> <tr> <td>muddy water</td> <td>troubled water</td> </tr> <tr> <td>rough wall</td> <td>shallow water</td> </tr> <tr> <td>damaged wall</td> <td>dirty bathroom</td> </tr> <tr> <td>ugly wall</td> <td>cold front</td> </tr> </tbody> </table>	cold water	dirty wall	muddy water	troubled water	rough wall	shallow water	damaged wall	dirty bathroom	ugly wall	cold front
cold water	dirty wall										
muddy water	troubled water										
rough wall	shallow water										
damaged wall	dirty bathroom										
ugly wall	cold front										

Figure 4.4: Mechanical Turk interfaces and instructions for *Collecting* sentences with a positive (top) and negative (bottom) sentiment.

4.4.3 Dataset Validation


	<p>The task is to rate how well each caption describes the image.</p> <p>If you think the sentiment (positiveness or negativeness) of the caption does not match the image tick the "wrong sentiment" checkbox.</p> <p>Scale guidelines follow:</p> <ol style="list-style-type: none"> Correctly describes the image <ul style="list-style-type: none"> Everything described in the sentence appears in the image. All the important parts of the image are described in the sentence The caption is allowed to describe things which you don't know are true (eg 'cold water' even if you cant tell the water is cold) Almost describes the image <ul style="list-style-type: none"> Major details described in the sentence appear in the image. Most of the important parts of the image are described in the sentence Barely describes the image <ul style="list-style-type: none"> Only some minor details described in the sentence appear in the image. Unrelated to image <ul style="list-style-type: none"> No details described in the sentence appear in the image. 																																			
Caption	How descriptive?																																			
	<table border="1"> <thead> <tr> <th>Correctly</th> <th>Almost</th> <th>Barely</th> <th>Unrelated</th> <th>Wrong Sentiment</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> <tr> <td><input type="radio"/> 1</td> <td><input type="radio"/> 2</td> <td><input type="radio"/> 3</td> <td><input type="radio"/> 4</td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	Correctly	Almost	Barely	Unrelated	Wrong Sentiment	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>
Correctly	Almost	Barely	Unrelated	Wrong Sentiment																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																
<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="checkbox"/>																																

Figure 4.5: Mechanical Turk interface and instructions for validating the dataset.

We use a further AMT task to evaluate the quality of the collected dataset and validate the plausibility of the sentiment captioning task. In particular, we look to answer two questions: is the descriptiveness of the caption affected by introducing sentiment?, Can we pair both positive and negative sentiment with a single image?

Our validation uses the two-question AMT task shown in Figure 4.5, applied to 124 images with 3 neutral captions from MSCOCO, and images with 3 positive and 3 negative captions from our dataset. We first ask AMT workers to rate the descriptiveness of a caption for a given image on a four-point scale [Hodosh et al., 2013; Vinyals et al., 2015b]. We also ask whether the sentiment of the sentence matches the image. Each rating task is completed by 3 different AMT workers.

Results in Figure 4.6 show descriptiveness tends to decrease when the caption contains additional sentiment – see the *descriptiveness* column. Ratings for the positive captions (Pos) exhibit a small decrease (by 0.08, or one-tenth of the standard deviation), while negative captions (NEG) exhibit a large decrease (by 0.73). Reviewing the data indicates that the smaller set of negative ANPs (436 negative ANPs vs 1027 positive ANPs) makes it hard for annotators to produce visually congruent sentiment. In general, this result shows sentiment can be added to a caption while retaining descriptiveness, although it is easier with a more flexible vocabulary.

In the *correct sentiment* column of Figure 4.6, we record the number of votes each caption received for bearing a sentiment that matches the image. We can see that the vast majority of the captions are unanimously considered emotionally appropriate

			#sents with <i>wrong sentiment</i> by #votes				
	#imgs	#sents	desc.	0 votes	1 vote	2 votes	3 votes
COCO	124	372	3.42±0.81	355	16	1	0
POS	124	335	3.34±0.79	325	20	0	0
NEG	123	305	2.69±1.11	250	49	6	0

Figure 4.6: Summary of quality validation for sentiment captions. The rows are MSCOCO [Chen et al., 2015], and captions with Positive and Negative sentiments, respectively. *Descriptiveness* \pm *standard deviation* is rated as 1–4 and averaged across different AMT workers, higher is better. The *Correct sentiment* column records the number of captions receiving 3, 2, 1, 0 votes for having a sentiment that matches the image, from three different AMT workers.

(94%, or 315/335 for Pos; 82%, or 250/305 for NEG). Among the captions with less than unanimous votes received, most of them (20 for Pos and 49 for NEG) still have majority agreement for having the correct sentiment, which is on par with the level of noise (16 for Coco captions). Frequently, images can be described with both positive and negative sentiment. We have the freedom to choose the sentiment polarity we wish to apply to the image caption pair.

Our results have important implications for stylistic caption generation beyond polarised sentiment. First, we have demonstrated that the a image can be appropriately described in more than one, apparently conflicting style. This suggests we have some freedom to choose the style of the caption. Second, the descriptiveness is adversely affected by a restrictive style space. With a broad sentiment style the problem is limited, but some stylistic goals may be so restrictive as to promote inconsistencies in image-caption pairs.

4.5 Experiments

Section 4.5.1 and Section 4.5.2 give details of the testing environment and settings, including hyper-parameters and datasets.

4.5.1 Implementation Details

We implement RNNs with LSTM units using the Theano package [Bastien et al., 2012]. Our implementation of CNN+RNN reproduces caption generation performance in recent work [Karpathy and Fei-Fei, 2015]. The visual input to the switching RNN is 4096-dimensional feature vector from the second last layer of the Oxford

VGG-16 CNN [Simonyan and Zisserman, 2015]. These features are linearly embedded into a $D = 512$ dimensional space. Our word embeddings \mathbf{E}_y , LSTM hidden state \mathbf{h} and cell memory \mathbf{c} also have 512 dimensions. The size of our vocabulary for generating sentences is 8,787, and becomes 8,811 after including additional sentiment words.

We train the model using Stochastic Gradient Descent (SGD) with mini-batching and the momentum update rule. Mini-batches consist of 128 examples, the momentum is fixed at 0.99, and the learning rate is fixed at 0.001. We clip gradient norms to the range $[-5, 5]$; a standard practice to prevent exploding gradients in LSTM training [Graves, 2013]. Training is complete when the perplexity fails to improve over ten consecutive check-points (defined as every fifth mini-batch). The entire system has approximately 48 million parameters, and learning them on the sentiment dataset takes about 20 minutes at 113 image-sentence pairs per second, while the original model on the MSCOCO dataset takes around 24 hours at 352 image-sentence pairs per second. Given a new image, we generate the caption with the highest length normalised likelihood using a beam-search of beam-size 5. A multi-core workstation with an Nvidia K40 GPU was used for all experiments.

4.5.2 Dataset Setup

The factual RNN is learned on the MSCOCO training set [Chen et al., 2015] of 413K+ sentences on 82K+ images. We construct an additional set of caption with sentiments as described in Section 4.4.2 using images from the MSCOCO validation partition. The Pos subset contains 2,873 positive sentences and 998 images for training, and another 2,019 sentences over 673 images for testing. The Neg subset contains 2,468 negative sentences and 997 images for training, and another 1,509 sentences over 503 images for testing. Each of the test images has three positive and/or three negative captions.

4.5.3 Baselines

Our first baseline is *CNN+RNN* an LSTM with CNN input [Vinyals et al., 2015b], trained on the sentiment neutral MSCOCO dataset. This is a state-of-the-art model for generating descriptive image captions. It also forms the basis of our SentiCap model. Comparison with this baseline allows us to judge the relative merits of our novel modifications.

Our next two baselines, *ANP-Replace* and *ANP-Scoring*, build on the aforementioned *CNN+RNN* baseline. Both modify sentences generated with an *CNN+RNN* by adding an adjective with strong sentiment to a random noun. *ANP-Replace* adds

the most common adjective, in the sentiment captions, for the chosen noun. *ANP-Scoring* uses multi-class logistic regression to select the most likely adjective for the chosen noun, given the Oxford VGG-16 features. This is equivalent to retraining the last layer of the Oxford VGG-16 network for adjective prediction, keeping all other weights fixed.

The next model, denoted as *RNN-Transfer*, learns a fine-tuned RNN on the sentiment dataset with additional regularisation from *CNN+RNN* [Schweikert et al., 2008], as in $R(\Theta)$ (cf. Eq (4.8)). Unlike SentiCap, this baseline is a single RNN stream.

We name the full switching RNN system as *SentiCap*, which jointly learns the RNN and the switching probability with word-level sentiments by Equation (4.7).


4.5.4 Evaluation Metrics

We evaluate our system both, with automatic metrics, and judgements crowd-sourced through Amazon Mechanical Turk. Automatic evaluation uses the BLEU, ROUGE_L, METEOR, CIDE_r metrics from the Microsoft COCO evaluation software [Chen et al., 2015]. The BLEU metric is a corpus-level n-gram precision score between the reference sentences and the candidate sentences. ROUGE_L is a longest common sub-sequence based f-measure, weighted to favour high recall. METEOR calculates the best reference-to-candidate alignment using exact matches, WordNet synonyms and stemming; the result is the harmonic mean of precision and recall between the best aligned match. CIDE_r includes tf-idf weights when scoring the n-gram matches. These metrics have become the standard for evaluating automatic image-caption generators, and are known to correlate with human evaluations [Vinyals et al., 2017]. For further details of these metrics see Section 2.3.4.

In our crowd-sourced evaluation task, AMT workers are given an image and two automatically generated captions displayed in a random order – the interface is shown in Figure 4.7. Each task consists of three different types of rating: most positive, most interesting and descriptiveness. The most positive and most interesting ratings are pair-wise comparisons, with one caption from the sentiment neutral *CNN+RNN* model, the other from *SentiCap* or a baseline. Descriptiveness is rated from 1-4 on a per-caption basis and aggregated by taking the mean. Caption pairs are rated by three different AMT workers; a caption is considered more positive/negative than its pair if at least two workers agree. There are 5 images per task – an essential component because of AMT’s pricing strategy.

We found that asking Turkers to rate sentences using this method initially produced very poor results, with many Turkers selecting random options without reading the sentences. We suspect bots were primarily to blame. Our first solution was

to use more skilled Turkers, called master workers. Although this lead to cleaner results, the smaller number of workers meant that a large batch of tasks took far too long to complete. Instead we used workers with a 95% or greater approval rating. To combat the quality issues we randomly interspersed the manual sentiment captions from our dataset, and then rejected all tasks from a worker who failed to achieve 60% accuracy for the most positive rating. This was an effective way of filtering out poor quality workers and bots. There were few cases where workers were close to the 60% accuracy cut-off; they were typically much higher or much lower than the threshold, which validates the idea that some workers were not completing the task correctly.



This HIT consists of 5 sets of 3 judgments. Click the next button to move to the next set of judgments. You must make all 3 judgments before you can move on.

The task is to make three judgments for each of the caption pairs which relate to the shown image.

- Which caption describes the image using the **most positive** (strongest positive sentiment) wording? (select the caption)
- In your opinion which is the **more interesting** caption of the two? (select the caption)
- How well do the captions describe the image? (Rate 1 to 4)
- If all the words in the sentences are identical select **Sentences are identical** (in this case you do not need to make the other judgments)

Scale guidelines:

- Correctly describes the image
 - All the important parts of the image are described in the sentence
 - The caption is allowed to describe things which you don't know are true (eg 'cold water' even if you cant tell the water is cold)
- Almost describes the image
 - Most of the important parts of the image are described in the sentence
- Barely describes the image
 - Only some minor details described in the sentence appear in the image.
- Unrelated to image
 - No details described in the sentence appear in the image.

75%

Caption	Most positive	More interesting	Describes the image			
			Correctly	Almost	Barely	Unrelated
a group of people on a boat in a body of water	1	1	1	2	3	4
a great group of people on a boat in the calm water	1	1	1	2	3	4
<input type="checkbox"/> Sentences are identical Next						

Submit

Figure 4.7: AMT interface and instructions for *comparative rating* of generated sentiment sentences

4.5.5 Results

Table 4.2 summarises the automatic evaluations. *SentiCap* produces substantially more captions with sentiment ANPs than any of the baseline methods, demonstrating an ability to consistently incorporate sentiment. In comparison, captions generated by *CNN+RNN* contain few sentiment ANPs, because it is trained on the primarily sentiment neutral MSCOCO dataset. That *SentiCap* generates more sentiment ANPs than the two insertion baselines *ANP-Replace* and *ANP-Scoring* shows *SentiCap* actively drives the flow of the sentence towards sentimental ANPs.

The automatic metrics measuring similarity to the ground truth (BLEU, ROUGE_L, METEOR, CIDE_r) show only small changes across methods. This is in-part because we

		SEN%	B-1	B-2	B-3	B-4	ROUGE _L	METEOR	CIDE _r
Pos	CNN+RNN	1.0	48.7	28.1	17.0	10.7	36.6	15.3	55.6
	ANP-Replace	90.3	48.2	27.8	16.4	10.1	36.6	16.5	55.2
	ANP-Scoring	90.3	48.3	27.9	16.6	10.1	36.5	16.6	55.4
	RNN-Transfer	86.5	49.3	29.5	17.9	10.9	37.2	17.0	54.1
	SentiCap	93.2	49.1	29.1	17.5	10.8	36.5	16.8	54.4
NEG	CNN+RNN	0.8	47.6	27.5	16.3	9.8	36.1	15.0	54.6
	ANP-Replace	85.5	48.1	28.8	17.7	10.9	36.3	16.0	56.5
	ANP-Scoring	85.5	47.9	28.7	17.7	11.1	36.2	16.0	57.1
	RNN-Transfer	73.4	47.8	29.0	18.7	12.1	36.7	16.2	55.9
	SentiCap	97.4	50.0	31.2	20.3	13.1	37.9	16.8	61.8

Table 4.2: Summary of automatic evaluations for captions with sentiment. Columns: SEN% is the percentage of output sentences with at least one ANP; B-1 ... CIDE_r are automatic metrics as described in Section 4.5; where B-N corresponds to the BLEU-N metric measuring the co-occurrences of n-grams.

only need to change a few important words to introduce sentiment, and since these metrics do not weight words by importance, measured changes are small. Moreover, to score well under these metrics a model must not only introduce sentiment at the right location, but also choose the same word as the human annotator. This is hard because, as we have shown in Section 4.4.3, there are multiple valid ways of conveying sentiment about an image. *SentiCap* shows a strong improvement in all automatic metrics for the negative sentences, which is because the number of valid ways to describe an image with negative sentiment is limited – this is seen by the smaller number of negative ANPs compared to positive ANPs and the reduced set of ANPs chosen by human annotators.

Table 4.3 presents the crowd-sourced evaluations. Sentences from *SentiCap* are, on average, judged by crowd sourced workers to have stronger sentiment than any of the three baselines. For positive *SentiCap*, 88.4% are judged to have a more positive sentiment than the CNN+RNN baseline. These gains are made with only a small reduction in the descriptiveness – further analysis shows this decrease is due to a minority of failure cases, since 84.6% of captions ranked favourably in the pair-wise descriptiveness comparison. Negative captions generated by *SentiCap* are judged to have greater negative sentiment 72.5% of the time, which is slightly weaker than the positive case. On the other hand automatic metrics, show *SentiCap* performed better at generating negative captions than positive ones, and outperforming all three baselines. As there are fewer new adjectives in the negative ANP set *SentiCap* is likely able to learn more reliable statistics, leading to greater performance on automatic metrics. In regard to human evaluation, it is unclear whether a cultural bias towards positive evaluation [Heine and Lehman, 1995; Sharot, 2012] came into play. Perhaps

		SENTI	DESC	DESCCMP
Pos	CNN+RNN	–	2.90±0.90	–
	ANP-Replace	84.8%	2.89±0.92	95.0%
	ANP-Scoring	84.8%	2.86±0.96	95.3%
	RNN-Transfer	84.2%	2.73±0.96	76.2%
	SentiCap	88.4%	2.86±0.97	84.6%
NEG	CNN+RNN	–	2.81±0.94	–
	ANP-Replace	61.4%	2.51±0.93	73.7%
	ANP-Scoring	64.5%	2.52±0.94	76.0%
	RNN-Transfer	68.1%	2.52±0.96	70.3%
	SentiCap	72.5%	2.40±0.89	65.0%

Table 4.3: Summary of crowd-sourced evaluations for captions with sentiment. Columns: SENTI is the fraction of images for which at least two AMT workers agree that it is the more positive/negative sentence; DESC contains the mean and std of the 4-point descriptiveness score: larger is better. DESCMP is the percentage of times the method was judged more descriptive, or equally descriptive, as the CNN+RNN baseline.

using workers from Japanese culture, which has been shown to be less optimistic than some western cultures [Heine and Lehman, 1995] would yield a more balanced result. In practice, this would be difficult due to differences in language and perceived sentiment [Jou et al., 2015]. We do not evaluate the effects of culture on sentiment evaluation, but rather leave it as an exciting area for future work.

SentiCap sentences with positive sentiment were judged by AMT workers as *more interesting* than those without sentiment in 66.4% of cases, which shows that our method improves the expressiveness of the image captions. On the other hand, negative sentences were judged to be *less interesting* than those without sentiment in 63.2% of cases. This is mostly due to negativity in the sentence being a natural contradiction to being *interesting*, a positive sentiment.

It has been noted by Vinyals et al. [2015b] that RNN captioning methods tend to exactly reproduce sentences from the training set. Our SENTICAP method produces a larger fraction of novel sentences than an RNN trained on a single caption domain. A sentence is novel if there is no match in the MSCOCO training set or the sentiment caption dataset. Overall, SENTICAP produces 95.7% novel captions; while CNN+RNN, which was trained only on MSCOCO, produces 38.2% novel captions – higher than the 20% observed by Vinyals et al. [2015b].

Constructing the sentiment dataset for *SentiCap* required a sentiment vocabulary described in Section 4.4.1. This vocabulary is fixed but relatively broad, with 322 unique nouns and 212 unique adjectives; however, the effective size of the vocabulary at generation time is much smaller. We find only 75 unique nouns and 71 unique

<i>Noun</i>	<i>Adjectives</i>
man	happy(56.7%), nice(39.4%), good(3.8%)
field	sunny(100.0%)
room	nice(97.7%), great(2.3%)
group	great(100.0%)
food	tasty(92.9%), healthy(4.8%), delicious(2.4%)
building	beautiful(95.1%), nice(4.9%)
street	busy(55.3%), nice(18.4%), beautiful(13.2%), calm(10.5%), pleasant(2.6%)
woman	beautiful(75.0%), pretty(25.0%)
cat	cute(56.7%), adorable(26.7%), cuddly(16.7%)
people	happy(39.1%), nice(34.8%), beautiful(21.7%), great(4.3%)

Table 4.4: ANPs for **positive** sentences generated by *SentiCap*. Nouns are ordered from most common to least, with only the ten most common shown. Paired adjectives are ordered most common (left) to least (right); only the five most common are shown. Percentages reflect the fraction of times the adjective was paired with the noun.

adjectives from the ANP vocabulary appear in the 1176 generated test captions (this includes 673 positive and 503 negative captions). In part we attribute this to the limited number of concepts in MSCOCO, the limited set of test captions, and annotator preferences for certain adjectives and nouns. However, this is not the full story as human annotators use a broader vocabulary: matching ground-truth test captions to the ANP vocabulary, and randomly down-sampling to 1176 captions (to match the number of generated captions), gives 143 unique adjectives and 173 unique nouns. This indicates that part of the problem is the dataset, but that effective vocabulary reduction is an inherent limitation of *SentiCap* which requires further study. This limitation would be most apparent when reviewing generating multiple captions for different images: they could become repetitive and lose their effectiveness. For isolated images we suspect the vocabulary limitations would be less apparent.

Table 4.4 shows the most common positive ANPs generated by *SentiCap*. The common nouns “*field*” and “*group*” only have one adjective, highlighting the limited diversity, e.g. not all “*fields*” are “*sunny*”. Other nouns such as “*man*”, “*street*”, and “*cat*” have multiple different adjectives. Table 4.5 shows the most common negative ANPs generated by *SentiCap*. Similar to the positive case, we see common nouns with only one adjective “*street*”, “*people*”, and “*bathroom*”. In this sample the generated adjectives are more restricted for negative sentiment than for positive sentiment – this observation holds in general, and is supported by ANP vocabulary counts (1027 positive ANPs, 436 negative ANPs).

Table 4.6 displays, for each method, the mean number of each POS class per gen-

<i>Noun</i>	<i>Adjectives</i>
man	dead(99.0%), invisible(1.0%)
street	lonely(100.0%)
building	ugly(83.8%), damaged(13.5%), abandoned(2.7%)
people	stupid(100.0%)
bathroom	dirty(100.0%)
food	disgusting(57.7%), bad(42.3%)
grass	dead(100.0%)
cat	lazy(45.5%), annoying(18.2%), stupid(13.6%), silly(13.6%), dirty(9.1%)
train	lonely(95.7%), abandoned(4.3%)
water	cold(36.4%), dirty(36.4%), shallow(9.1%), troubled(9.1%), muddy(9.1%)

Table 4.5: ANPs for **negative** sentences generated by *SentiCap*. Nouns are ordered from most common to least, with only the ten most common shown. Paired adjectives are ordered most common (left) to least (right); only the five most common are shown. Percentages reflect the fraction of times the adjective was paired with the noun.

erated **positive** sentiment sentence. *SentiCap* and all sentiment producing baselines generate longer sentences with substantially more adjectives than the purely descriptive *CNN+RNN* method. *ANP-Replace* and *ANP-Scoring* produce, on average 0.9, more adjectives per sentence than *CNN+RNN* and add on average 0.9 more words; this is by construction as these baselines add an extra adjective where possible. *SentiCap* introduces slightly fewer adjectives than these direct replacement baselines – at an additional 0.83 per sentence – but produces sentences that are on average only 0.28 words longer than *CNN+RNN*. This indicates *SentiCap* largely, but not exclusively, functions as a sophisticated adjective insertion method. However, unlike the baseline insertion methods *SentiCap* does not increase the sentence length in proportion to the number of adjectives added. This is primarily achieved by generating fewer noun phrases per sentence, as measured by noun phrase chunking [Ramshaw and Marcus, 1999] implemented in spaCy¹: *SentiCap* produces 3.10, *CNN+RNN* produces 3.26, and *ANP-Replace* produces 3.26. This implies more focus on generating sentiment rather than describing all aspects of the scene. Similar conclusions can be drawn in the negative sentiment case, which is summarised in Table 4.6.

Figure 4.8 contains a number of example image captions generated by SENTICAP – the left half are positive, the right half negative. The highlighted text shows cases where the switch variable gives a high probability to the word being part of a sentiment ANP. We can see that the switch variable captures almost all sentiment phrases,

¹<https://github.com/explosion/spaCy/tree/v1.9.0>

<i>Model</i>	<i>DET</i>	<i>NOUN</i>	<i>ADP</i>	<i>ADJ</i>	<i>VERB</i>	<i>ADV</i>	<i>Total</i>
CNN+RNN	2.61	3.81	2.13	0.60	1.06	0.07	11.47
ANP-Replace	2.61	3.80	2.09	1.50	1.07	0.07	12.37
ANP-Scoring	2.61	3.80	2.10	1.49	1.08	0.08	12.37
RNN-Transfer	2.58	3.74	1.83	1.52	1.28	0.07	12.23
SentiCap	2.55	3.53	1.78	1.43	1.06	0.24	11.75

Table 4.6: The mean number of each POS class per sentence for the **positive** sentiment generated sentences. Note *CNN+RNN* generates MSCOCO style captions all other methods generate **positive** sentiment.

<i>Model</i>	<i>DET</i>	<i>NOUN</i>	<i>ADP</i>	<i>ADJ</i>	<i>VERB</i>	<i>ADV</i>	<i>Total</i>
CNN+RNN	2.60	3.75	2.10	0.62	1.07	0.08	11.42
ANP-Replace	2.60	3.76	2.06	1.45	1.09	0.08	12.27
ANP-Scoring	2.60	3.75	2.09	1.43	1.11	0.08	12.27
RNN-Transfer	2.61	3.83	2.11	1.33	1.26	0.11	12.43
SentiCap	2.51	3.50	1.78	1.37	1.12	0.04	11.52

Table 4.7: The mean number of each POS class per sentence for the **negative** sentiment generated sentences. Note *CNN+RNN* generates MSCOCO style captions all other methods generate **negative** sentiment.

and some of the surrounding words (e.g. “train station”, “plate”). Examples in the first two rows are generally descriptive and accurate such as “delicious piece of cake” (2a), “ugly car” and “abandoned buildings” (1c). Results for the other examples contain more or less inappropriateness in either the content description or sentiment, or both. (3b) captures the “happy” spirit correctly, but the semantic of a child in playground is mistaken with that of a man on a skateboard due to very high visual resemblance. (3d) interestingly juxtaposed the positive ANP “clever trick” and negative ANP “dead man”, creating an impossible yet amusing caption.

4.6 Summary

This chapter proposed SentiCap, a switching RNN model for generating image captions with sentiments. One novel feature of this model is a specialised word-level supervision scheme to effectively make use of a small amount of training data with sentiments. Also designed was a crowd-sourced caption re-writing task for generating descriptive captions with sentiment. I demonstrate the effectiveness of the proposed model using both automatic and crowd-sourced evaluations, with the SentiCap model able to generate an emotional caption for over 90% of the images. The

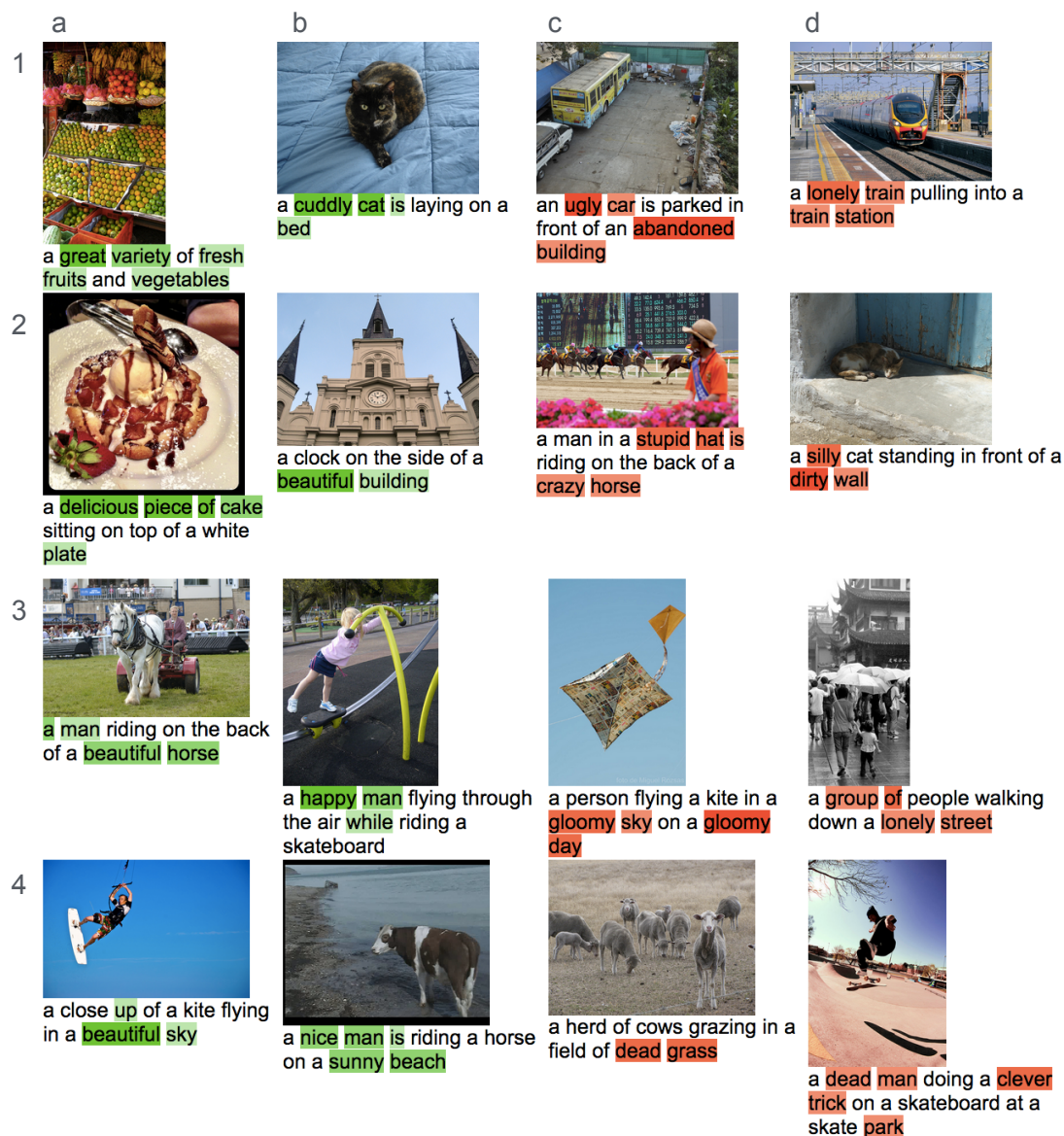


Figure 4.8: Example results from sentiment caption generation. Columns a+b: **positive** captions; columns c+d: **negative** captions. Background colour indicates the probability of the switching variable $\gamma_t^1 = p(s_t|\cdot)$: **dark** if $\gamma_t^1 \geq 0.75$; **medium** if $\gamma_t^1 \geq 0.5$; **light** if $\gamma_t^1 \geq 0.25$. Examples in rows 1 and 2 are successful. Examples in rows 3 and 4 have various semantics or sentiment errors, at times with amusing effects. See Section 4.5 for discussions.

vast majority of the generated captions were rated by crowd workers as having the appropriate sentiment.

The approach used by SentiCap is more broadly applicable to generating image captions with a stylistic component. SentiCap learns to balance two conditional

language models, one tuned for semantics, the other for style. Switching between these two models occurs on a word-to-word basis, giving rise to expression of a well defined style and coherent semantics. Styles which can be presented with simple atomic components – such as ANPs – are most suited to this approach. For example sentiment, regional slang or different levels of object description specificity, all fit comfortably in the SentiCap framework. Unfortunately, more complex styles that require global sentence changes, including word re-ordering (eg language simplification, literary style), do not fit naturally into this framework – more subtle techniques are required, such as those I present in Chapter 5 and Chapter 6.

The SentiCap approach has a fixed trade-off between style and semantics, set at training time by the data and hyper-parameters of the word-level regularizer. Ideally this trade-off would be a test time parameter, making pre-trained models more portable; it is unclear how to enable a test-time trade-off given the importance of the annotator defined trade-off implicit in the data.

SentiCap overcomes data scarcity by using transfer learning from large set of neutral image captions to a small set of stylistic captions. Word level regularization contributes to this success by incorporating word level sentiment information from external sentiment corpora. This approach substantially reduces the cost – both time and money – of building stylistic caption generators, making them more applicable in a range of scenarios. However, the barrier to entry is still high; for each style we require human annotations over the same image domain as the neutral captions. In Chapter 6 I introduce a novel styled image captioning model that works without being trained on styled captions.

The proposed data collection method, which guides annotators in sentence re-writing, allows the collection of semantically relevant styled sentences from untrained annotators. This method is most applicable for styles described by atomic components, such as ANPs. Moving to styles that are not well described by atomic components would be more challenging by would not require a complete paradigm shift: supervised learning could still be used. Completely new approaches are needed to deal with the case where style is difficult to describe to annotators, such as the style of a literary genera or particular author.

SentiCap shows that it is possible to change the style of an image caption with limited effect on the descriptiveness; however, when the style space is limited the descriptiveness can suffer – as seen by the negative sentiment experiments. This becomes an important consideration when attempting to generate more specific and therefore more restrictive styles. We will lose descriptiveness unless we also increase the range of changes the style can impose. Rather than just ANP replacement we may also need word re-ordering, phrase insertion or phrase deletion.

Images can be paired appropriately with both positive and negative sentiment captions. This supports the hypothesis that the style of the caption is not tied to the image and that we have some degree of choice. Although this was not shown in the general style case, the fact that it holds with styles as strongly polarised as sentiment is promising.

Simplifying Sentences

5.1 Introduction

In this chapter I focus on language modelling techniques for shifting linguistic style without modifying semantics. Working only with text, I develop techniques for ameliorating the data scarcity problem: the limited number of sentences of one style aligned to another style. By exploiting pre-training and properties specific to style translation, I work on generating sentences that are both semantically relevant and stylistically expressive.

Specifically, I explore the task of rewriting sentences to make the language easier to understand while preserving semantics. This task cannot be considered entirely about style generation as content removal or addition (eg inline definitions) may be necessary. Nevertheless, the use of simple, understandable language can certainly be considered an aspect of style, notably used by the prolific author Ernest Hemmingway [Müller, 2009].

Texts come in different levels of complexity, from technical pieces written for domain experts to simple books for children. Automated text simplification has the potential to allow readers to quickly understand information outside their specific background. The benefits will be even greater for new language learners or people with language impairments. Automatic simplification could adapt complex texts for a large audience, reduce misinformation and aid information flow between different cultures and technical disciplines. I tackle an important sub-problem – text simplification by sentence rewriting, for example, simplifying the sentence “*Not many foods inspire a fandom quite like Nutella.*” to “*Not many foods excite fans quite like Nutella.*” – a simplification requiring phrase replacement. This is in contrast to similar tasks such as lexical simplification [Specia and Jauhar, 2012; Paetzold and Specia, 2016], sentence compression [Cohn and Lapata, 2009; Rush et al., 2015], and text summarization [Cheng and Lapata, 2016; Nallapati et al., 2017].

Sentence simplification requires complex re-writes such as phrase replacement,

sentence restructuring or splitting [Cohn and Lapata, 2008; Zhu et al., 2010]. For example, the complex sentence “*A scientist has found the first wild alligator snapping turtle spotted in Illinois in more than 30 years.*” could be simplified to “*Wild alligator snapping turtles are hard to find in Illinois. They have not been seen in years. Until now.*”. Techniques developed for simplification could be employed to generate other types of style text requiring complex re-writes. In Chapter 3, I explored word replacements and in Chapter 4 I explored small free-form rewrites that were mostly adjective replacements or insertions. This chapter works towards richer style inclusion. Linguistic simplification is also useful in its own right, for language learning, information dispersal and text accessibility. Another factor in the decision to explore linguistic simplification is the availability of data. Recently a high quality – but somewhat limited in size – dataset became available [Xu et al., 2015b].

I adapt neural Sequence to Sequence models [Sutskever et al., 2014; Luong et al., 2015] for the Sentence Simplification problem, to develop a new model called S4 (Sequence to Sequence for Sentence Simplification). There are three novel components to S4: a large vocabulary with both pre-trained and learned word embeddings mitigates the effects of limited training data; a word-copy feeding algorithm exploits linguistic similarities between the original and simplified sentences with the help of an attention mechanism; a novel loss function encourages word-copying, ensuring the output sentences benefit from the rich input vocabulary despite having limited training data. The combination of these components can be seen as trading off semantic relevance and style expression. Copied text is semantically relevant but not necessarily consistent with the target style, while generated words should be style appropriate. The copying mechanism means the new word generator can focus on generating style specific text, and avoid wasting capacity on shared linguistic properties.

Sentences generated by S4 are simpler than the input, and preserve the original meaning. Compared to reference sentences, word-copying with the novel objective improves BLEU-4 by 4.9 points, and using the right mixture of pre-trained and learnt embeddings leads to a further 3.8 point improvement.

5.2 Related Work

Sentence simplification sits within a set of re-writing tasks, including: machine translation, lexical simplification, sentence compression, and summarization. However, none of the aforementioned re-writing tasks are solved, nor are they drop in solutions to sentence simplification, so they can only guide an approach. In section 5.2.1 I examine previous attempts at using machine translation for sentence simplification.

In section 5.2.2 I consider lexical simplification, sentence compression, and summarization. Section 5.2.3 explores the datasets available for sentence simplification.

5.2.1 Sentence Simplification as Machine Translation

Machine translation is similar to sentence simplification, although machine translation is a more developed area with many system designs already having been thoroughly explored. This makes machine translation an excellent source of inspiration for sentence simplification. There have been several attempts to adapt machine translation to sentence simplification. We group these by method as: phrase translation [Coster and Kauchak, 2011; Wubben et al., 2012; Stajner and Saggion, 2015], parse tree translation [Zhu et al., 2010; Woodsend and Lapata, 2011], and external paraphrase corpora [Xu et al., 2016; Pavlick and Callison-Burch, 2016].

Phrase-based machine translation [Och et al., 1999; Marcu and Wong, 2002; Koehn et al., 2007] is a class of statistical translation model [Lopez and Adam, 2008]. Statistical translation models normally have two parts: a translation model and a language model. The translation model gives the probability of the source language text given the target language text, while the language model gives the prior probability of the target language text. These two components are combined using Bayes rule during text generation (called decoding) to give the probability of the target given the source. The most likely output sentence can then be found by searching through the space of output sentences, a task that can be non-trivial if the translation or language models are not designed to allow efficient decoding. Decoding with phrase-based models has the additional step of segmenting the input sentence into phrases. Learning the parameters of the translation model requires paired sentences in the source and target language. Word level alignments can be approximately recovered using the IBM alignment models [Brown et al., 1993], which recover alignments by modelling translation as set of source to target generation steps. These steps vary depending on the specific model but can include, expanding source words in to many target words (one-to-many alignment), introducing new words (aligning to null), translating words individually, and localised re-ordering of words. Individual word translations can be learnt by counting (or soft counting) aligned words – these then feed back into the alignment model. Phrase translations can be learnt by searching through the aligned words for frequently aligned sub-strings, this can be constrained using syntactic parsers for both languages [Koehn et al., 2003].

Phrase-based machine translation is a common approach to sentence simplification [Wubben et al., 2012; Stajner and Saggion, 2015] in part because of open source tools such as Moses [Koehn et al., 2007]. Wubben et al. [2012] use Moses to gener-

ate a short list of candidates that they re-rank by levenshtein distance to the input. Stajner and Saggion [2015] evaluate the effect of training data size and quality on simplifications generated by Moses. Both groups show that phrase-based machine translation outperforms some simple baselines but does not consistently outperform the unmodified input text as judged by machine translation metrics.

Another class of approaches to machine translation is parse tree translation, where a source-language parse tree is transformed into a target-language parse tree [Lopez and Adam, 2008]. This is designed to incorporate knowledge of language structure into the translation task and may help when translating languages with very different word order (eg English to Japanese). Yamada and Knight [2001] develop a model for parse tree translation based on stochastic operations at each node of a constituency parse tree. The permitted operations are: reordering child nodes, for which a single probability table of all possible reordering is learnt; inserting extra words, for which constituent insertion probability and word insertion probability tables are learnt; and translating leaf words, for which a translation table is learnt. The learning process involves using the EM algorithm to maximise the likelihood on the training corpus. An alternative approach to parse tree translation is Synchronous Context-Free Grammars (SCFGs): a variant of context free grammars when the output is two strings, namely the source and target sentences [Wu and Hong Kong, 1995; Lopez and Adam, 2008]. For example, while a context-free grammar might have a production rule from a noun phrase to a noun phrase and an adjective $NP \rightarrow NP JJ$, a SCFG could have a rule from a noun phrase to a noun phrase and adjective in two different orders $NP \rightarrow NP JJ/JJ NP$, one for each language. To learn a SCFG for translation EM can be used, first pairs of parsed sentences in both languages are parsed under the grammar simultaneously perhaps using top-down dynamic programming (though other approaches may be used if the particular SCFGs cannot be applied efficiently via this method [Gildea and Satta, 2016]), next the model parameters are updated based on this dual parse. Decoding the model simply requires parsing a single input sentence using the first half of each production rule, though additional syntactic or linguistic knowledge may be applied on the output language by incorporating language or syntactic model likelihood.

Parse tree approaches to sentence simplification have also shown some promise. Zhu et al. [2010] propose a tree-based sentence simplification method that uses the constituency parse tree to guide sentence splitting, word dropping, word reordering, and word or phrase substitution. To decide when to perform these actions, they learn probability tables with simple features using the sentence aligned simple wikipedia dataset. For example, when deciding where to split a sentence, they use: the word surface form, the word constituent (parse tree label), and integer normalised sen-

tence length. Word substitution is likewise based on learnt translation probabilities. The full model produces shorter sentences that are also simpler sentences as judged by Flesch (reading ease metric), although the Moses baseline obtains a better BLEU score. Woodsend and Lapata [2011] also implement a tree-based sentence simplification method; they break down sentences into component phrases and clauses, simplify each using a translation grammar, before constructing the simplified sentence using an Integer Linear Programming (ILP) formulation. The translation grammar, a Quasi-synchronous grammar [Smith and Eisner, 2006], is a generalisation of SCFGs introduced previously. It is learnt using simple wikipedia revision history and sentences aligned between simple wikipedia and standard english wikipedia. The ILP formulation includes grammatical constraints, while a linear approximation to the Flesch score helps to form the objective. This model outperforms the earlier model of Zhu et al. [2010] in human evaluations of grammaticality and meaning preservation.

External paraphrase corpora are an attractive option for sentence simplification, as they can substantially reduce the training data requirements. The most common choice of external paraphrase corpus is PPDB [Ganitkevitch et al., 2013]: a large publicly released corpus, built using machine translation phrase alignments. Mono-lingual paraphrases are extracted by identifying common phrase alignments from English to another language, such as Spanish, and then alignments from Spanish back to English. As PPDB paraphrases are not necessarily simpler, Pavlick and Callison-Burch [2016] attempt to identify simplifying PPDB paraphrases. First they select a set of common paraphrases from PPDB and collect crowd-source judgements on whether they are: simplifying, complicating, or incorrect paraphrases. Using these judgements as training data, they employ a linear classifier to annotate the remaining PPDB paraphrases. Xu et al. [2016] build a sentence simplification pipeline base on PPDB. They use a linear model implemented as part of the Moses machine translation toolkit to choose the most appropriate simplification, based on features such as the length in characters, number of syllables, and language model score. The resulting Moses model outperforms the more direct Moses model of Wubben et al. [2012], which does not use an external paraphrase corpora.

5.2.2 Related Problems

Lexical simplification is a sub-problem of sentence simplification, involving the replacement of a word or n-gram with a simpler alternative – re-ordering or deletion are not permitted. The problem can be broken down into complex word identification [Paetzold and Specia, 2016], and substitution selection [Specia and Jauhar, 2012].

Paetzold and Specia [2015] summarise a range of feature based approaches [Szarvas et al., 2013; Horn et al., 2014] and develop a modular toolkit named LEXenstein that tackles both subtasks. From the surveyed approaches, they conclude that the most accurate approach is to identify complex words with a binary classifier, select possible substitutions with word2vec [Mikolov et al., 2013], and then choose the most appropriate with a binary classifier. The classifiers use a large number of features including: word morphology, n-gram probabilities and document frequencies. More recent work shows embeddings from bi-directional LSTMs outperform word2vec similarity [Melamud et al., 2016] for substitution selection. While the effect of LSTMs on the entire lexical simplification pipeline has yet to be explored, this result does suggest that LSTMs are capable of capturing the broader semantic context necessary for simplification.

Sentence compression involves reducing the length of a sentence by removing phrases, while retaining grammatical correctness and the original meaning. This task targets short output sentences, with no requirement that they are simpler. Although, the resulting sentences will be shorter and are likely to be syntactically simpler, complex words may be used as a way of conveying information compactly. Previous solutions relied on external corpora and parse trees [Jing, 2000; Cohn and Lapata, 2009]. More recently, large parallel corpora [Filippova and Altun, 2013] have led to interest in end-to-end learning [Filippova et al., 2015; Rush et al., 2015; Auli and Rush, 2016]. Jing [2000] presents a sentence compression model that parses sentences into tree structures and annotates each sub-tree with: grammatical importance (as defined by external corpora and linguistic rules), connection to local context (as defined by local word repetitions and similar word repetitions), and likelihood of being removed by human annotators (as defined by a small text corpora). Phrases are then removed in a top down fashion by thresholding the annotation scores. An alternative approach by Cohn and Lapata [2009] uses a large margin method to learn weights for a synchronous tree-substitution grammar on a training set of a few thousand sentences. Their tree-to-tree re-writing technique outperformed the state-of-the-art (in 2009), and is applicable to rewriting problems beyond word-deletion, such as sentence simplification. More recently, Filippova et al. [2015] use a neural network encoder-decoder model (see Section 2.2.2) to tackle sentence compression. They train end-to-end on a parallel corpus of 2 million sentences built from news article headlines and first sentences [Filippova and Altun, 2013]. Their model beats the state-of-the-art approach in automatic and human evaluations. Other authors [Rush et al., 2015; Auli and Rush, 2016] extend this model to abstractive compression, where generated words are not a strict subset of the original sentence.

Text summarization involves taking a long text as input and outputting a much

shorter text that captures the most salient information in the original text. A typical summary marks a trade-off between including information and making the text shorter, it does not necessarily cover the entire content of original text. The summaries of interest here are those written in natural language, thus language fluency must also be traded-off against information density. To complicate matters further, different summarization variants have different definitions of salient information. Variants of summarization include multi-document summarization (summarising an entire document collection), update summarization (summarising what is new in a collection, for example new news topics), and personalised summarization (customised summarization based on user preferences or queries). For a comprehensive review of text summarization and its variants see Lloret and Palomar [2012]. Methods for summarization fall into two broad categories [Mani et al., 2002]. Extractive (or surface-oriented) approaches select and join segments of the original text, while abstractive (or knowledge-rich) approaches extract an intermediate semantic representation that is later used to generate entirely new sentences. To identify important text segments, surface-oriented approaches use many different properties: characteristic word cues [Edmundson, 1969] (eg “*in summary ...*”), frequency of non-stop words [Luhn, 1958; McCargar, 2004; Lloret and Palomar, 2009], probability under a topic model, graph based connectivity estimates [Erkan and Radev, 2004; Mihalcea and Tarau, 2004] (where text segments are nodes and edges represent text segment similarity), and sentence classification and re-ranking [Li et al., 2007; Wong et al., 2008a]. Knowledge-rich approaches often employ a knowledge base specific to a particular domain, such as: biomedicine [Fiszman et al., 2004; Rindflesch et al., 2011], patents [Wanner et al., 2008], or toy domains [Moawad and Aref, 2012].

The more recent approaches to summarization have employed deep neural networks and learnt intermediate representations – often referred to in this context as embeddings or hidden states. Enabling the use of deep learning models, which require large amounts of data, is the CNN/DailyMail corpus [Hermann et al., 2015] with human written summaries for almost 300,000 news articles. The first use of this dataset for summarization was by Cheng and Lapata [2016]. They employ an encoder with a two level hierarchy: a word level CNN embeds words in each sentence, while an RNN combines these into a document level embedding. Their best performing model generates summaries by classifying input sentences – using the learnt embedding – as either part of the summary or not part of the summary. This is a sequence labelling problem for which they employ an LSTM. Nallapati et al. [2017] take a similar approach, but use bi-directional GRUs to combine word embeddings for each sentence. A separate bi-directional GRU accepts these sentence embeddings and classifies each as either part of the summary or not part of the summary. To

enable training of their extractive model on abstractive summaries, a GRU decoder is connected to a weighted sum of all sentence embeddings. Training this decoder to minimise the log probability of words in the summary is claimed to help learn better sentence embeddings. The decoder is only used during training. Automatic evaluations put this model on par with the model of Cheng and Lapata [2016]. See et al. [2017] use an encoder-decoder model for abstractive summarization. Their encoder and decoder are both LSTMs with the decoder outputting three different probability distributions at each time step: P_{gen} a Bernoulli distribution for word generation vs copying, P_{vocab} a Categorical distribution over output words in the decoder vocabulary, and P_{attn} a Categorical distribution over input words (the attention component). The final probability of outputting word w is:

$$P_{final}(w) = P_{gen}P_{vocab}(w) + (1 - P_{gen}) \sum_{i:w_i=w} P_{attn}(i) \quad (5.1)$$

The loss function is average log likelihood with a coverage penalty to reduce the chance of repeatedly attending to the same words. In terms of ROUGE and METEOR score, their method is weaker than a baseline that selects the first 3 sentence of the article, and the best extractive only model [Nallapati et al., 2017]. However, they outperform previous state-of-the-art abstractive methods and demonstrate the output contains novel sequences.

5.2.3 Datasets

Many recent attempts at sentence simplification [Zhu et al., 2010; Coster and Kauchak, 2011; Horn et al., 2014] use the simple wikipedia dataset [Zhu et al., 2010]. This dataset was constructed by aligning sentences from paired articles in English Wikipedia and Simple English Wikipedia¹. The Simple English Wikipedia is written by volunteers in a similar way to English Wikipedia, though they are encouraged to use only the 1000 most common English words, simple grammar, and shorter sentences. These are not strictly enforced, but rather considered broad guidelines. For example, using words outside the 1000 most common is permitted, and relatively frequent in practice. The simple wikipedia dataset consists of 108,016 paired sentences extracted from 65,133 articles; the average sentence length is 25.01 in wikipedia and 20.87 in simple wikipedia.

Xu et al. [2015b] recently showed that simple wikipedia dataset contains a large number of inadequate simplifications and is prone to sentence alignment errors. They instead suggest the Newsela dataset, sourced (with permission) from the online news source Newsela², which consists of news articles re-written by professional

¹https://simple.wikipedia.org/wiki/Main_Page

²<https://newsela.com/>

editors to target different reading grades. These are roughly aligned with grades 3, 4, 6, 7 and 12, under the Common Core Standards in the United States. A thorough analysis by Xu et al. [2015b] shows that compared to simple wikipedia, Newsela has a more consistent level of quality with a higher degree of simplification. They estimate that only 50% of sentences in simple wikipedia are true simplifications, while at least 90% of Newsela sentence pairs are true simplifications. The number of true simplifications increases to 92% when only considering alignments between the most complex articles and the most simple articles.

5.3 Model

Section 5.3.1 provides an overview of the neural sequence-to-sequence model and its encoder and decoder components, followed by three novel components of S4 - mixing pre-trained and trainable word embeddings (Section 5.3.2), word-copy feeding (Section 5.3.3), and a custom loss function (Section 5.3.4).

We denote the inputs to the encoder and decoder as \mathbf{x}^{enc} and \mathbf{x}^{dec} , the outputs words as \mathbf{y} , the attention vector for the i 'th output token as \mathbf{a}_i , and the sequence state vectors as \mathbf{h}_j^{enc} and \mathbf{h}_i^{dec} . The encoder sequence is indexed by j ; the decoder sequence by i . The encoder sequence length is M and the decoder sequence length is L . We use bold-face for vectors and upper-case for matrices.

5.3.1 Sequence to Sequence with Attention

Our base sequence to sequence model (Figure 5.1) uses two sets of Gated Recurrent Units (GRUs) [Cho et al., 2014a,b]. The encoder GRU embeds the sentence into a set of vectors, while the decoder GRU generates text from this set of vector embeddings. GRUs are a popular Recurrent Neural Network (RNN) that perform similarly [Chung et al., 2014] to the Long Short Term Memory (LSTM). For more information on GRUs see Section 2.2.1; for more information on sequence to sequence models see 2.2.2. The last hidden output of our encoder GRU is transformed by a fully connected linear layer and then input to the decoder GRU as the first hidden state. Both GRUs have two layers, each with 512 units, and act on sentences of up to 50 words. The 300 dimensional word embedding matrices E^{enc} , E^{dec} are linearly projected into the 512 dimensional input space.

We implement the global attention model from Luong et al. [2015] that was originally designed for machine translation. Attention is a short circuit from the sequence encoder to the sequence decoder output – for more details on attention models see Section 2.2.2.1. In our formulation, the attention vector \mathbf{a}_i for the i 'th output token is

calculated as the softmax $\sigma(z)$ over inner products of the current decoder state with each of the encoder state vectors.

$$\mathbf{a}_i = \sigma(\mathbf{h}_{0:M}^{enc} \cdot (\mathbf{h}_i^{dec})^T) \quad (5.2)$$

$$\sigma(z) = \frac{e^z}{\sum_{j=0}^M e^{z_j}} \quad (5.3)$$

The resulting attention \mathbf{a}_i weights the output of the encoder, which forms the context vector \mathbf{c}_i .

$$\mathbf{c}_i = \sum_{j=0}^M \mathbf{a}_{i,j} \mathbf{h}_j^{enc} \quad (5.4)$$

The context vector is concatenated with the decoder output and input to a feed forward layer with learnt parameters W^{out} and softmax non-linearity. The output is the distribution over the next word $p(y_i | x^{enc}, x_{0:i}^{dec})$.

$$p(y_i | x^{enc}, x_{0:i}^{dec}) = \sigma(W^{out}[\mathbf{c}_i, \mathbf{h}_i^{dec}]) \quad (5.5)$$

Where $[\mathbf{c}_i, \mathbf{h}_i^{dec}]$ denotes concatenation of the context and decoder hidden vectors to form a new vector.

We train end-to-end using dropout [Srivastava et al., 2014], mini-batched adaptive gradient descent algorithm Adam [Kingma and Ba, 2015], and early stopping. Dropout was applied to: the word projection layer output, the encoder hidden outputs, the context vector, and the decoder output. The dropout ratio was set to 0.7, we found that such a large value (0.5 is more usual) helped to prevent over-fitting given our relatively small dataset and large numbers of learn-able parameters. For Adam, the learning rate was set to 0.001, β_1 was 0.9 and β_2 was 0.999 – β_1, β_2 are exponential decay rates for the first and second moment estimates. Note that Adam is typically insensitive to the chosen hyper-parameters [Kingma and Ba, 2015]. The mini-batch size was 256 sentence pairs and the score on 1024 validation samples was used for early stopping.

5.3.2 Mixing Pre-trained and Trainable Word Embeddings

A large vocabulary is necessary to represent complex sentences; however, as we show in Section 5.5.2, learning embeddings for a large vocabulary when training data is limited can hurt performance. Instead we extend the size of the input vocabulary with pre-trained GloVe [Pennington et al., 2014] embeddings. Specifically, we learn embeddings for the 5000 most frequent words, and use fixed GloVe embeddings for an additional 640,317 words. The number of learnt embeddings was chosen with grid

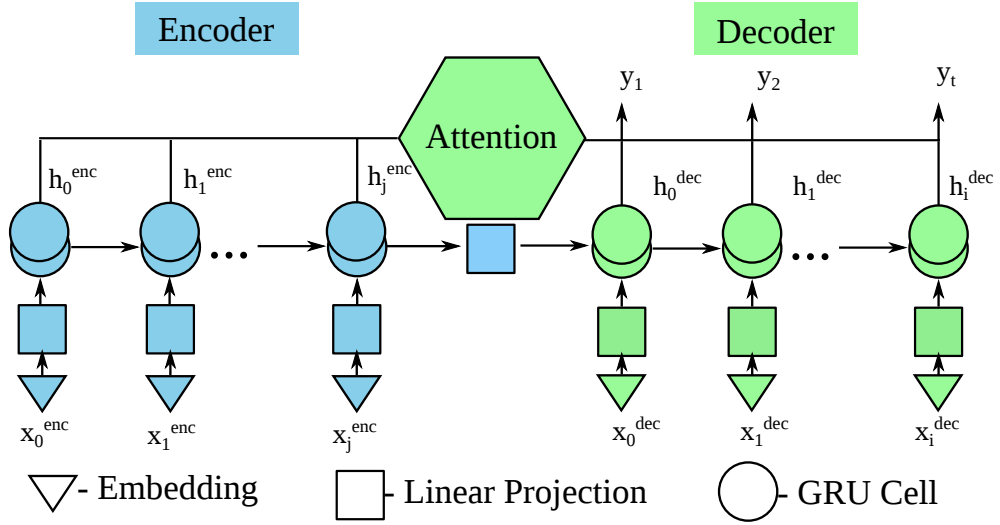


Figure 5.1: The encoder-decoder with attention for sentence simplification.

search. Words without an embedding (learnt or GloVe) are replaced with a shared UNK token. These results are included in Section 5.5.2.

When the dataset covers a large range of different topics – such as news articles – words not seen, or infrequently seen during training, may still be frequent in the test set. Pre-trained word embeddings can help to cope with this disconnect; however, using only a few learnt embeddings leads to a low variance model that cannot fit the training data effectively. By choosing a mixture of pre-trained and trainable embeddings we balance these two objectives. The learnt embeddings are restricted to the more frequent words as these have the most training data.

Extending the input vocabulary does not increase the computational cost, because we only learn embeddings for the most frequent words. Unfortunately, we cannot extend the output vocabulary without significantly increasing the computational cost of the final softmax, which is already the most expensive component for model training.

5.3.3 Attentive Word-Copy Feeding

We design an attentive word-copy feeding mechanism to copy rare words that are absent from the output vocabulary but are in the input vocabulary. This also takes advantage of the similarity between the input and output sentences. A special output token *cpy* is introduced to denote a copy operation. When generated at position i , we copy the word $x_{j^*}^{enc}$ from the input that is the most likely alignment, computed

by attention score as $j^* = \underset{j=0:M}{\operatorname{argmax}}\{a_{i,j}\}$, with encoded sentence length denoted M .

This technique has been used in machine translation to deal with limited vocabulary sizes [Luong et al., 2014], but has only been applied during post-processing. In order to take advantage of a larger input vocabulary, we feed the copied word – rather than the *cpy* token itself – as input x_{i+1}^{dec} in the next step of sentence generation. Feeding the copied word allows the model to see more of the final sequence, which improves performance when paired with our loss function that encourages copying. This is especially important in the case of simplification, where a large proportion of words are copied rather than generated as in the machine translation case.

5.3.4 Loss Function for Word-Copying

We designed a two-part loss function (Eq 5.6) to take advantage of the similarities between the simplified and original sentences. The first term is the categorical cross-entropy, a common loss function for encoder-decoder models [Sutskever et al., 2014; Luong et al., 2015], minimising it increases the probability of generating the ground truth word \hat{y}_i from the softmax output. The second term is a binary cross-entropy (Eq 5.7). It encourages word copying at each position i when the input word with the maximum attention $x_{j^*}^{enc}$ agrees with the correct ground-truth word \hat{y}_i . Intuitively, the model learns when direct copying of the input is appropriate. We first train with the categorical cross-entropy and then fine-tune with the two-part loss function. Where \mathbb{I} denotes the indicator function.

$$\mathcal{L} = \frac{1}{L} \left(- \sum_{i=0}^L \log P(y_i = \hat{y}_i) + bce(\hat{y}, y) \right) \quad (5.6)$$

$$\begin{aligned} bce(\hat{y}, y) = \sum_{i=0}^L & - \log P(y_i = cpy)^{\mathbb{I}(x_{j^*}^{enc} = \hat{y}_i)} \\ & - \log(1 - P(y_i = cpy))^{\mathbb{I}(x_{j^*}^{enc} \neq \hat{y}_i)} \end{aligned} \quad (5.7)$$

5.4 Evaluation Settings

5.4.1 Newsela Dataset

The Newsela dataset [Xu et al., 2015b] is a collection of English news article sets, where each article set consists of a source article at 5 levels of simplification. The source article (considered year 12 level) was rewritten by professional linguists for (approximate) grades 3, 4, 6, 7 under the Common Core Standards in the United States. The Newsela dataset is only available upon request to Newsela, and the set

of articles provided has not been standardised – they simply provide all published articles up to the time of the request. We received 1,911 article sets, while the initial report by Xu et al. [2015b] consisted of 1,130 article sets. We use all 1,911 article sets to ensure maximum training data – all our baselines use this full dataset.

The Newsela articles are grouped into sets, but the sentences are not aligned. We align sentences (as per Section 5.4.1.1) from the 4 most complex levels to the most simple level. This choice ensures we have an approximately fixed simplification target, and a large number of aligned sentences. While other alignment options are possible, we found them to be more difficult to learn.

We split the Newsela dataset into 1337 training article sets, 191 validation article sets and 383 test article sets. We remove identical aligned sentence pairs, leaving: 105,917 training sentences, 15,858 validation sentences and 28,468 test sentences. Because we split by article rather than sentence, there is a vocabulary difference between training and testing, making our setting more challenging.

For S4 the input vocabulary has 645,317 words. For baselines which do not exploit pre-trained embeddings, the input vocabulary constructed from Newsela has 31,630 words. In all cases the output vocabulary is restricted to the most frequent 10,000 words that occur at least 7 times in training. The size of the output vocabulary is not crucial because word copying ensures any word in the input vocabulary can end up in the simplified sentence.

5.4.1.1 **Aligning Sentences**

The Newsela dataset in its raw form is aligned only at the document level. We align the different rewrites at the sentence level using a dynamic programming algorithm loosely based on the work of Coster and Kauchak [2011]. Our approach allows sentence splitting, where two simple sentences align to a single complex sentence. We also take sentence ordering into account, which permits a low similarity score threshold, which increases the number of matches.

The main components of any dynamic programming algorithm are the sub-problems, which break the task into manageable chunks, and the recurrence relationship, which describes how to combine solutions to sub-problems. Our sub-problem, denoted $a(i, j)$, is the optimal score for aligning all sentences in the complex document after and including index i to all sentences in the simple document after and including index j . The recurrence relation that describes how to build up these sub-problems is defined in Equations 5.8-5.11. First, we define s_i as the i 'th complex sentence and s_j as the j 'th simplified sentence. Note that in this section, i and j denote sentence indices rather than word indices as was the case in Section 5.3.4. The

similarity function between two sentences is denoted $d_{i,j}$ and defined in Equation 5.9. D_{comp} is the number of sentences in the complex document, and M_i is the number of words in the i 'th complex sentence. D_{simp} is the number of sentences in the simplified document, and L_j is the number of words in the j 'th simplified sentence.

For clarity we present the recurrence relationship in two parts: the first matches each complex sentence with a single simple sentence; the second matches each complex sentence with two simplified sentences.

For single sentence matching there are three possible cases reflected in Equation 5.8. If we choose to match the sentences, the score is the similarity of the two sentences $d_{i,j}$, plus the best score for all later alignments $a(i+1, j+1)$. If we choose not to match the two sentences, the score is γ (the skip penalty) plus the best score for all later alignments: this is $a(i+1, j)$ if we skip the complex sentence and $a(i, j+1)$ if we skip the simple sentence.

$$a^{single}(i, j) = \max(a(i+1, j+1) + d_{i,j}, a(i+1, j) + \gamma, a(i, j+1) + \gamma) \quad (5.8)$$

The sentence similarity function is defined in terms of the BLEU-4 score as:

$$\begin{aligned} \sigma(s_i, s_j) &= \text{BLEU-4}(s_i, s_j) \\ d_{i,j} &= \min(\sigma(s_i, s_j), \sigma(s_j, s_i)) \end{aligned} \quad (5.9)$$

We use BLEU-4 because its sensitivity extends up to a four-gram overlap, but also includes tri-gram, bi-gram and uni-gram overlap. We found BLEU-4 gave reasonable alignments in most cases. Note that BLEU-4 varies between 0 and 100, with 100 being the highest similarity. For more details on how BLEU is computed see Section 2.3.4.

For multi-sentence matches we consider splitting the complex sentence into two parts. The recurrence is described in Equation 5.10. Here p is the split index for the complex sentence, with each fragment aligned to a different simplified sentence. Since we consider local sentence re-ordering, there are two options for each index p . The first option has the complex sentence prefix aligned to the first simple sentence and the suffix aligned to the second simple sentence. The second option has the prefix aligned to the second sentence and the suffix aligned to the first sentence. In both cases we add the best score for all later alignments $a(i+1, j+2)$.

$$\begin{aligned} a^{multi}(i, j) = \max_{p < D_{comp}} \max(\sigma(s_{i,[1:p]}, s_j) + \sigma(s_{i,[p:M]}, s_{j+1}), \\ \sigma(s_{i,[1:p]}, s_{j+1}) + \sigma(s_{i,[p:M]}, s_j)) + a(i+1, j+2) \end{aligned} \quad (5.10)$$

The notation $[\alpha : \beta]$ denotes all integer values between α and β , inclusive. Using Equation 5.10 and Equation 5.8 we define the full recurrence as:

$$a(i, j) = \begin{cases} 0 & \text{if } i \geq D_{simp} \\ 0 & \text{if } j \geq D_{comp} \\ \max(a^{single}(i, j), a^{multi}(i, j)) & \text{otherwise} \end{cases} \quad (5.11)$$

As is usual in a dynamic programming algorithm this recurrence is efficiently computable when caching sub-problems. Once the optimal alignment score is found, the alignments can be recovered by backtracking through the cache and choosing the action at each i, j position that lead to the optimal score.

5.4.1.2 Manual Word Level Alignment

To evaluate the attention mechanism, we chose a subset of 512 sentence-pairs from the validation set and created a ground-truth alignment at the word level. An automatic matching approach based on longest contiguous matching subsequence (commonly known as diff) made an initial set of matches. One annotator – the author of this thesis – then reviewed the sentence-pairs and corrected all misaligned or missing alignments.

5.4.2 Moses Baseline

Many authors [Coster and Kauchak, 2011; Wubben et al., 2012; Stajner and Saggion, 2015] have applied the open source phrase translation software *Moses* [Koehn et al., 2007] to sentence simplification. We adopt *Moses* as a baseline which we train by following the directions of Coster and Kauchak [2011]. Specifically, we keep the default settings for tokenization and truecasing, remove sentences longer than 80 words, and train a tri-gram language model using modified Kneser-Ney smoothing. The hyperparameters are tuned with Minimum Error Rate Training (MERT), which maximises the BLEU score on a sample of 400 paired sentences from the validation set. Using a small sample from the validation set is necessary because the MERT algorithm is computationally expensive; our sample size is consistent with Coster and Kauchak [2011].

5.4.3 Evaluation metrics

We use three types of metrics that measure: the similarity, the amount of change, and the simplicity of the generated sentences. The similarity metrics BLEU [Papineni et al., 2002] ($B1-B4$) and *Rouge* [Lin, 2004] are commonly used for evaluating machine translation – larger scores mean greater similarity to the ground-truth. For a detailed discussion of BLEU and Rouge see Section 2.3.4. The distance to the original sentence

	B-1	B-2	B-3	B-4	Rouge	Flesch	Avg.Words	Edit Dist.
ground-truth	-	-	-	-	-	74.69	15.72	7.30
original	69.84	62.76	57.57	53.10	75.07	64.75	17.12	0.0
moses	65.43	56.45	49.94	44.50	69.99	74.19	17.08	1.56
S4-attn	23.35	13.54	8.77	5.95	30.69	91.68	9.70	13.46
S4-feed	61.94	51.94	45.14	39.71	64.75	75.49	15.49	5.94
S4+gv+bce-feed	16.86	7.70	3.52	1.72	24.42	67.91	27.54	26.08
S4	63.04	53.60	46.96	41.51	65.91	77.90	15.26	5.53
S4+gv	67.51	59.01	52.90	47.75	70.28	73.41	15.34	3.82
S4+bce	65.28	57.23	51.36	46.43	68.03	74.72	15.94	4.70
S4+gv+bce	68.71	60.80	55.11	50.28	71.02	68.71	15.80	3.51

Table 5.1: Results for the end-to-end sentence simplification task. Our complete model is *S4+gv+bce*. Section 5.4.1 details the metrics. Values in bold are closest to the ground-truth. For BLEU or Rouge the largest value is closest to the ground-truth, for Flesch or Average Words the closest has the smallest delta from a Flesch of 74.69 or a Average Words of 15.72.

is measured by edit distance (*Edit Dist.*), the number of word insertions, deletions or substitutions to turn the original sentence into the generated sentence. Sentence simplicity is measured by average words per sentence (*Avg.Words*) and Flesch-Kincaid reading ease (*Flesch*). *Flesch* score is a widely used open-source metric for simplification tasks [Zhu et al., 2010; Narayan and Gardent, 2014]. It weights average words per sentence and average syllables per word – simpler sentences have higher scores.

5.5 Results

Table 5.1 summarises the performances of the model variants and baselines. We use suffixes to show the components added or removed to each *S4* model variant: *-attn* for removing attention, *-feed* for removing word-copy, *+gv* for allowing a mix of trainable and pre-trained embeddings, and *+bce* for training with the loss function for word-copying. The base model for *S4* is an encoder-decoder model with attention and word-copy feeding.

5.5.1 Sequence to Sequence Performance

Our model *S4+gv+bce* outperforms phrase translation trained with open source software *Moses* [Koehn et al., 2007] and used by earlier simplification work [Coster and Kauchak, 2011; Wubben et al., 2012; Stajner and Saggion, 2015]. The sentences generated by *S4+gv+bce* have higher BLEU and Rouge scores than *Moses*, indicating greater similarity to the simplified sentences. The *Moses* baseline achieves a good Flesch score; however, coupled with the lack of similarity to the simplified sentences, this could indicate a loss of semantics.

We find that the *original* sentences achieve high BLEU and Rouge scores despite being longer and more complex than the generated sentences – this is consistent with previous observations [Coster and Kauchak, 2011; Wubben et al., 2012; Stajner and Saggion, 2015]. The dataset construction is partially responsible: using BLEU-4 for aligning ground truth sentences (see Section 5.4.1.1) means there is a large overlap between original and simplified sentences. The content domain is also responsible as there is not much change between complex sentences and those simplified by linguists. For example, in the manually aligned section of the dataset only 6.1% of aligned words were changed going from the complex to simple sentences – the remaining 93.9% were copied. It is also likely that the BLEU and Rouge metrics evaluation have a hand in the high performance of the *original* sentences. These metrics penalise sentences shorter than the ground-truth – BLEU explicitly and Rouge implicitly. Generated simplifications will be penalised more frequently as they tend to be shorter than the *original* sentences.

5.5.2 Ablation Study

Each component in the *S4* model contributes to the performance, as shown in Table 5.1. Removing either the attention (*S4-attn*) or the feeding (*S4-feed*) causes a drop in BLEU and Rouge, indicating that the generated sentences are further from the simplified ground-truth. The attention is the more important of the two, with removal leading to an enormous 35.56 BLEU-4 point drop – we also observed semantic divergence from the input sentence after three to four words. Attention not only enables word copying but also reduces the effective depth of the network and avoids compressing the entire encoded sentence into a single fixed size vector. Adding either the pre-trained word-vectors or the custom loss function improves the BLEU and Rouge scores. *S4+bce* has a higher BLEU-4 by 4.9 points, and *S4+gv+bce* leads to a further 3.8 increase. Compared to the ground-truth sentences, *S4+gv+bce* is closest in sentence length, while *S4+bce* is closest in Flesch score.

The custom loss and word-copy feeding are designed to be paired. If we keep the custom loss function but remove feeding (*S4+gv+bce-feed*), performance degrades drastically, because the *cpy* token, which provides little information by itself, is used frequently – 87.7% (up from 9.0%).

By extending the vocabulary with pre-trained word-vectors (*S4+gv*), we mitigate the affects of data scarcity. Figure 5.2 shows that using 5000 trainable embeddings gives the best validation performance. If instead we use too many fixed embeddings (left of Figure 5.2), the model lacks the flexibility necessary to learn accurate simplifications; likewise, too many trainable embeddings means the model lacks the

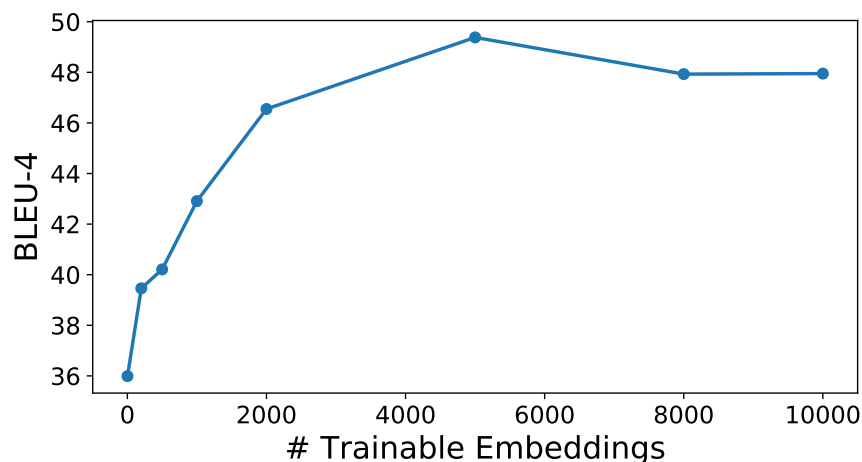


Figure 5.2: The validation performance of *S4+gv+bce* with different numbers of trainable and pre-trained embeddings. Higher BLEU-4 scores indicate greater simplification precision.

information to deal with uncommon words (right of Figure 5.2). This is a classic bias vs variance trade-off.

5.5.3 Simplification Examples

Table 5.2 shows two examples from the *S4+gv+bce* model. The first has “*massive*” changed to “*huge*”, which is semantically correct and subjectively simpler, and “*grain*” changed to “*vegetable*”, syntactically correct but a semantic mistake. The second example correctly splits a noun clause into two sentences, but the ground truth sentence uses additional information from article context, and is not shorter. These examples demonstrate that *S4* can learn to perform multiple simplification operations, including word replacement, sub-string removal and sentence splitting. They also illustrate the limitations of the available ground-truth. Having only one ground-truth simplification per original sentence can lead to penalising correctly simplified outputs. Moreover, the dataset was constructed with additional article level and local context not provided to the sentence simplifier.

5.5.4 Attention Alignment Performance

A direct evaluation of the attention alignments is presented in Table 5.3. To separate the effects of attention and word copying from imperfect word substitutions, we employ a word substitution oracle which always outputs the correct word from the ground-truth at test time. However, the correct word is only output if the alignment

<i>Orig</i>	The increase would put massive strains on the world’s water and grain supplies.
<i>Simp</i>	The increase would put huge strains on the world’s water and vegetable supplies.
<i>GT</i>	The increase would strain the world’s water and grain supplies.
<i>Orig</i>	Obama, who has not said when he’ll make a final decision, is under heavy pressure to approve the project.
<i>Simp</i>	Obama has not said when he’ll make a decision. he is under heavy pressure to approve the project.
<i>GT</i>	President Obama has not said when he’ll decide what to do about the pipeline. but, he is under pressure to say yes to the project.

Table 5.2: Simplification examples from the $S4+gv+bce$ model. Bold-face highlights changes from the original complex sentence to the simplified sentence. Insertions and replacements are bold-face in the simplified sentence, while deletions are bold-face in the complex sentence. *Orig* is the original complex sentence, *Simp* is the output of the $S4+gv+bce$ model, and *GT* is the ground truth simplified sentence.

	B-1	B-4	Rouge
S4	80.91	65.09	82.81
S4+bce	79.49	63.01	81.27
S4+gv	83.16	68.81	84.77
S4+gv+bce	81.92	66.83	83.50

Table 5.3: BLEU and Rouge scores when an oracle word simplifier is used: when the correct alignment is made the chosen word is guaranteed to be correct. This measures the performance of the attention layer alone.

given by the argmax attention is in the hand-aligned ground-truth (dataset details in Section 5.4.1). Words unaligned in the ground truth may have any argmax alignment.

Compared to Table 5.1, having a word replacement oracle boosts $S4$ BLEU-4 by 23.58 points and Rouge by 16.9 points – setting a theoretical upper bound on performance with the current attention model. The base model $S4$ with oracle achieves 65.1 in BLEU-4, while the model variants using pre-trained word embeddings ($+gv$) perform slightly better, with $S4+gv$ at 68.8 in BLEU-4. This demonstrates that both attention and word replacement components have a margin for improvement – future work focusing on either one would not be wasted effort.

5.5.5 Word Replacement Performance

We examine the performance of word replacement in isolation from the attention by using the hand-aligned data (Section 5.4.1.2). The goal of the decoder becomes com-

Generated	Ground Truth	
	Copy Word	Change Word
Copy Word	5359	259
Change Word	227	106

Table 5.4: Confusion matrix for choosing to change or copy a word. The rows are the actions chosen by the $S4+gv+bce$ model when fed ground-truth alignments. The columns are the ground truth actions.

puting the most likely next word given: the original sentence, all previously generated words, and the known alignment for the next word. To incorporate ground-truth alignments into the model, we replace the attention $a_{i,j}$ with the count normalised ground-truth alignments $\hat{a}_{i,j} = \frac{a_{i,j}^{gt}}{\sum_{m=0}^L a_{i,m}^{gt}}$. Where $a_{i,j}^{gt}$ is a ground-truth alignment indicator, with unit value if the j 'th input word is aligned to the i 'th output word, zero otherwise. Normalisation is necessary because multi-word alignments are permitted, and occur frequently in practice.

Table 5.4 is the confusion matrix for $S4+gv+bce$, showing changed words (words undergoing substitution) and copied words (words copied directly from the input). $S4+gv+bce$ frequently chooses to copy words, mirroring the high similarity between ground-truth sentences. However, in only 32% of cases does $S4+gv+bce$ correctly choose to change a word. Even when $S4+gv+bce$ correctly decides to change a word, it only chooses the same word as the ground-truth 46% of the time. Word replacement itself does not perform well. Better word replacement may be achieved with significantly more training data, though new ideas seem necessary for further improvement in the more common case of data scarcity.

5.6 Summary

I present S4, a sequence-to-sequence model for simplifying sentences. The new loss function encourages word copying, reducing the requirements on the word generator, and thereby narrowing the models to focus to the changes necessary for simplification. Word-copy feeding ensures the model sees an accurate word history even when copying is used extensively. The tune-able method for incorporating pre-trained word embeddings into the pipeline allows efficient use of external data – although much work in this area remains.

The remaining obstacles include low reliability of word substitutions, and the lack of aligned data. Future work includes exploiting datasets from related tasks. Attention alignment performance may also benefit from the coverage mechanisms

used in some recent sequence to sequence models [Tu et al., 2016; See and Manning, 2017].

Although this chapter deals entirely with text, the techniques developed are more broadly applicable to generating visually relevant styled captions, for example generating image captions with existing techniques, and then using the proposed style shifting model. This sidesteps the issue of finding image-caption pairs in the target style; however, since semantic pairs of sentences in different styles are scarce, this does not solve the data scarcity problem. In Chapter 6, I build on the idea of using sequence-to-sequence models for styled caption generation, but take tackling the data scarcity problem further by learning from an unaligned style corpus.

Captioning Images with Style Transfer from Unaligned Text Corpora

6.1 Introduction

In this chapter I address three main challenges. The first is human-like style transfer – using large amounts of unrelated text in a given style to compose styled image captions. The second is designing an intermediate space that separates the semantics from the linguistic style. The third is ensuring generated stylistic text remains descriptive and relevant to the image.

I develop a model, called *SemStyle*, for generating stylistically interesting and semantically relevant image captions by learning from a large corpus of stylised text without aligned images. Descriptive image captioning models are not applicable to this case because images are not aligned with styled captions. Central to my approach is separating semantic relevance and style. To this end, I propose a novel semantic terms representation that is concise and promotes flexibility in word choice.

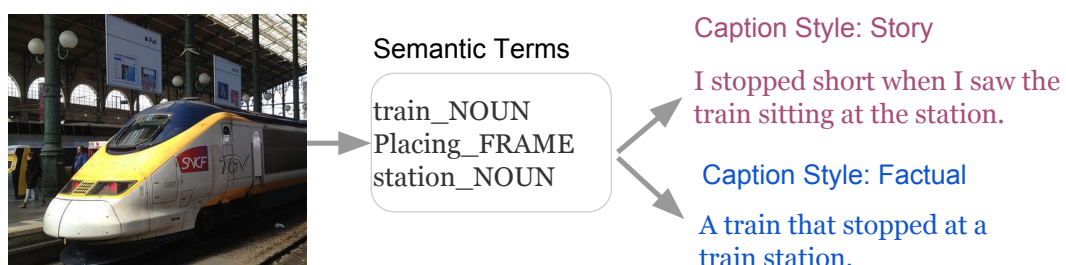


Figure 6.1: *SemStyle* distills an image into a set of semantic terms, which are then used to form captions of different styles.

Terms consist of normalised nouns with their part-of-speech tags, along with verbs generalised with the lexical database FrameNet. Further, I develop a *term generator* for obtaining a list of terms related to an image, and a *language generator* that decodes the ordered set of semantic terms into a stylised sentence. The *term generator* is trained on images and terms derived from factual captions. The *language generator* is trained on sentence collections and conditioned to generate the desired style. As illustrated in Figure 6.1, the *term generator* produces train_NOUN, Placing_Frame, station_NOUN from the image, and *language generator* produces sentences of different styles from this set of terms.

Evaluated on both MSCOCO [Chen et al., 2015] and a corpus of romance novels [Zhu et al., 2015], the SemStyle system produced distinctively styled captions in 58.8% of cases, while retaining visual semantics as judged by the SPICE metric [Anderson et al., 2016], and producing fluent sentences in the target style as judged by two different language models. Evaluated subjectively by the crowd, SemStyle achieved an average descriptiveness of 2.97 (out of 4, where larger is more descriptive) which is competitive with state-of-the-art purely descriptive baseline at 2.95. Since this baseline is the underlying method used by the semantic *term generator* component I demonstrate *SemStyle* does not significantly reduce caption relevance with respect to its underlying semantic model. Moreover, 41.9% of captions from SemStyle were judged to be telling a story about the associated image. The main contributions of this chapter are as follows:

- A concise semantic term representation for image and language semantics, implemented with a neural-network based *term generator*.
- A system for generating stylistic image captions without paired training data, using the semantic *term generator* and a stylistic *language generator*.
- A comparison with multi-modal vector space models.
- Competitive results in human and automatic evaluations with existing, and two novel, automated metrics for style.

6.2 Model

We propose a novel encoder-decoder model for generating semantically relevant styled captions. First, this model maps the image into a semantic term representation via the *term generator*, then the *language generator* uses these terms to generate a caption in the target style. This is illustrated in Figure 6.2.

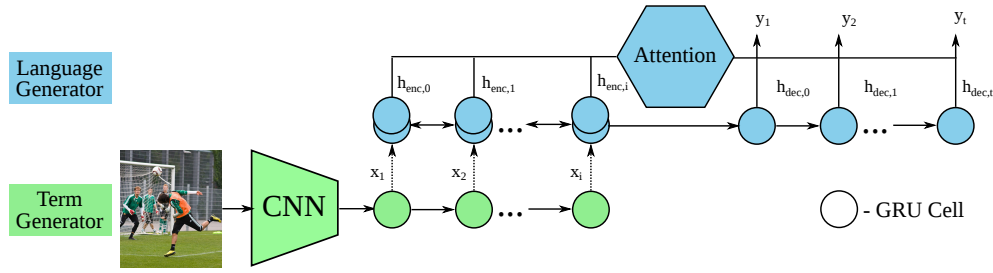


Figure 6.2: An overview of the *SemStyle* model. The *term generator* network (in green) is shown in the lower left. The *language generator* network is in the upper right (in blue) .

The lower left of Figure 6.2 describes the *term generator*, which takes an image as input, extracts features using a CNN (Convolutional Neural Network), and then generates an ordered term sequence, summarising the image semantics. The upper right of Figure 6.2 describes the *language generator*, which takes the term sequence as input, encodes it with an RNN (Recurrent Neural Network) and then, using an attention based RNN, decodes it into natural language with a specific style. We design a two-stage learning strategy enabling us to learn the *term generator* network using only a standard image caption dataset, such as MSCOCO [Chen et al., 2015], and learn the *language generator* network on styled text data, such as romantic novels.

The remainder of this section introduces our semantic representation and encoder-decoder neural network, while the learning method is discussed in Section 6.3.

6.2.1 Semantic Term Representation

To generate image captions that are both semantically relevant and appropriately styled, our structured semantic representation should capture visual semantics and be independent of linguistic style. When extracted from an image by the *term generator*, the representation should describe the contents of the image relevant to generating captions. We also need a representation that can be extracted from text alone so it can be used to train the *language generator* without images. Ideally, when extracted from text the semantic recall should be high to prevent the *language generator* from learning to invent semantics. However, this needs to be balanced with freedom for the *language generator* to choose language constructs that are stylistic in nature. In Section 6.5, we show our semantic term representation preserves the majority of real-world image and text semantics, while allowing the freedom to introduce style.

We opt to use discreet terms to form our semantic term representation, as it has some major advantages. Language itself is a discreet term space capable of accurately describing a huge variety of concepts; by staying close to a linguistic representation

we are able to describe a diversity of concepts. This also enables us to use ideas from the field of Natural Language Processing (NLP) – which has long been associated with extracting meaning from text. As the styled text should not effect the identification and selection of concepts from images we do not need to compute gradients from the *language generator* back to the *term generator*, allowing us to easily employ a discreet representation. In contrast a vector representation for terms has two main drawbacks. First, it would require learning a space that separates content and style, an unsolved task that is especially challenging in the case of romance novels and image captions which have substantially different content as well as style. Second, vector representations typically allow for greater variability in the final text realisation, whereas we aim to give the *term generator* direct control over the content words. Our discreet term space is a shared domain which naturally encompasses both the content of styled text and images.

Formally, given a sentence $\mathbf{w} = \{w_1, w_2, \dots, w_n\}$ with $w_i \in \mathcal{V}^{in}$, we define a set of rules mapping it to our ordered semantic terms $\mathbf{x} = \{x_1, x_2, \dots, x_M\}$, $x_i \in \mathcal{V}^{term}$. Our goal is to define a set of semantic terms and mapping rules broad enough to encompass the semantics of both images and stylistic texts, and yet specific enough to avoid encoding style. Inspired by computational stylistics we construct three sets of rules:

A. Filtering non-semantic words. Function words are known to encode style rather than semantics, and are often used in authorship identification models [Argamon and Levitan, 2005; Van Halteren et al., 2005; Argamon et al., 2003]. Here we remove function words in order to encode semantics and strip out style. From input sentence s , we filter English stop-words and a small list of additional terms, either informal e.g. “nah”, the result of tokenization e.g. “nt” or numbers e.g. “one”, “two”. Using Parts Of Speech (POS) tags, we further remove: punctuation, adverbs, adjectives, pronouns and conjunctions. The importance ordering of POS types is derived from a data-driven perplexity evaluation described in Section 6.2.2. Throughout this process we preserve the common collocation “hot dog”. This collocation was manually specified for our data set, but automatic approaches [Wehrli et al., 2010] could also be used to identify a more extensive list automatically. In most cases collocations are preserved without special exceptions, in the case of “hot dog”, the word “hot” was tagged as an adjective and so was filtered out.

B. Lemmatization and Tagging. Words from a sentence are converted to semantic terms to remove common surface variations. For most words we choose to lemmatize and concatenate with the POS (Part-Of-Speech) tag, e.g. “rock” becomes “rock_NOUN”. Lemmatization allows terms to be used more freely by the *language generator*, enabling stylistic choices such as tense and active/passive voice. POS tags

distinguish among different senses of the same word, for example the verb “rock” and the noun “rock” are disparate. We use the spaCy¹ natural language toolkit for lemmatization and POS tagging – the POS tagging method is based on Collins [2002].

C. Verb abstraction. To fully separate style and semantics it is necessary to remove word surface forms and reduce them down to semantic tokens. For example the words “walk”, “stroll”, and “stride” all have similar semantic content but different surface forms, where the choice of surface form is primarily stylistic. We rely on the concept of frame semantics when removing verb surface forms.

Frame semantics [Baker et al., 1998] is a conceptual framework for understanding semantics in language, where word meaning is based on contextual information called the semantic frame. This contextual information can come from surrounding words, in which case they are said to evoke a frame. For example the frame *Apply_heat* could be evoked by the words “fry”, “bake” and “boil”, and help us to distinguish the agent as a *Cook* and the subject as the *Food*. A large lexical database called FrameNet [Baker et al., 1998] defines over 1200 semantic frames and more than 13000 word senses. FrameNet provides the basis for a number [Johansson and Nugues, 2007; Das et al., 2014; Roth and Lapata, 2015; Swayamdipta et al., 2017] of frame-semantic role labelling techniques: where text is automatically annotated with semantic frames. A recent state-of-the-art approach named SEMAFOR [Das et al., 2014; Kshirsagar et al., 2015] is trained – via supervised and unsupervised techniques – to identify frame evocations, and collectively predict all arguments. We can use frame-semantic role labelling as a principled way of separating content from style.

We replace verbs with FrameNet [Baker et al., 1998] frames, preserving much of the semantics without enforcing a particular word choice. Table 6.1 displays the five most common verb frames in both the MSCOCO and Romantic Novel datasets. For instance, the lemmas: *sitting*, *laying* and *parking* all map to the *Placing* semantic frame. We use the semantic role labelling tool SEMAFOR [Kshirsagar et al., 2015] for sense disambiguation and frame annotation. We then map these frames into a reduced frame vocabulary, consisting of frames that occur over 200 times in the MSCOCO training set. Out-of-vocabulary frames are mapped to an in-vocabulary ancestor via the FrameNet hierarchy. Failing this, the frame is filtered out. Intuitively, frames not occurring frequently in the MSCOCO set, and with no frequent ancestors, are unlikely to be visually grounded – for example the frame *Certainty* with word lemmas “believe” and “trust” is a frame with no obvious visual grounding.

We choose not to use frames for other parts of speech as they are too broad, eg “dog”, “cat” and “cow” all map to the *Animal* frame and some common nouns

¹<https://github.com/explosion/spaCy/tree/v1.9.0>

were frequently labelled incorrectly e.g. “grass” labelled as an *Intoxicant*. We expect retraining SEMAFOR on visually grounded text such as image captions rather than books, letters and news articles would improve accuracy – though to our knowledge no such annotated dataset exists.

We do not explicitly determine word senses (other than for verbs), instead we rely on the *language generator* network to implicitly identify sense based on the context of other terms. Our early attempts to use word sense disambiguation explicitly were not fruitful due the tested systems making a large number of mistakes on the caption dataset. We found that our system works relatively well without explicit disambiguation, except in cases where there is a strong prior imposed by the styled text on the sense, for example a character in a romance novel called “Cat” could lead to confusion with the animal “cat”. To accurately use word sense disambiguation for styled image captioning it seems necessary to consider sense priors over the source and target domains, this is left as future work.

We preserve the ordering of the semantic terms from the original sentences, because results (Section 6.5) show ordering helps performance. For example, given the original sentence “A train that is stopped at a station.”, then the ordered set of semantic terms is “train_NOUN, placing_FRAME, station_NOUN”, where “placing_FRAME” is the frame for “stopped”. This allows the representation to distinguish between sentences such as “The dog bit the man.” and the “The man bit the dog.”.

SEMAFOR provides semantic roles for parts of the sentence which relate to the verb frame, for example “a train” could be annotated with the role *Theme* while “at a station” could be annotated by *Goal*. We do not use this role information in our system because it tends to be noisy (the incorrect roles are assigned) and sparse (there are a large number of possible role object pairs). Moreover, our method for mapping infrequent verb frames to more general and more frequent frames means that the roles relating to the original frame would appear out of their original context. However, across many different sentences terms with similar roles tend to be located in the same place relative to the verb, so retaining the ordering of semantic terms will tend to preserve role information. This is not universally true, consider the two sentences “The man who was bitten by the dog.” and “The man bit the dog.” which have the same term ordering but reversed roles. Designing a structured term representation that can differentiate between these cases, while being relatively dense and noise free is left as future work.

Frame (count)	Common Verbs
<i>MSCOCO Dataset</i>	
Placing (86,262)	sitting, parked, laying, hanging, leaning
Posture (45,150)	standing, lying, seated, kneeling, bends
Containing (32,040)	holding, holds, held, hold
Motion (22,378)	flying, going, swinging, fly, floating
Self motion (21,118)	walking, walks, walk, swimming
<i>Romantic Novels Dataset</i>	
Arriving (33781)	get, got, came, reached, come
Intentionally_act (33659)	do, did, does, doing, done
Self_motion (32911)	walked, walking, stepped, walk, slipped
Statement (32771)	said, told, say, says, talking
Placing (28986)	sitting, leaned, parked, hanging, placed

Table 6.1: The most common frames in the MSCOCO (596K training captions) and Romantic Novels Dataset (578K training sentences) with their frequency count and most common verbs.

6.2.2 Importance Ordering for Parts-of-Speech

We defined the set of semantic terms by incorporating our domain knowledge, e.g. nouns are semantically important while determiners are not. Alternatively, we can learn which word classes carry semantic information.

We would like to know which word classes (adjectives, nouns, verbs , etc.) carry the most visually semantic information per occurrence. To do this we seek the word classes which, when removed, lead to the largest increase in entropy after balancing for their frequency. The implicit assumption here is that semantic words more new information than style words and are therefore both the most difficult to predict and the most useful for predicting other words in the sentence. In practice we calculate the contribution of POS classes using the perplexity of the ground truth sentence after conditioning on input words belonging to different classes. For example, remove all nouns from the conditioning set of semantic terms and measure the change in perplexity. Balancing for class frequency is necessary to avoid bias towards the most frequent words.

It is not clear how to directly balance for class in this setup, instead we insure all conditioning sentences have a fixed fraction of their original number of words. For example, when removing nouns we also remove other types of words uniformly at random until the conditioning sentence has reached the desired length. This ensures that the perplexity score captures the predictability of words in the POS class and the importance of the POS class for predicting other words, rather than the number of occurrences.

Our approach requires a probabilistic model with a domain including the word classes of interest and a range including possible output sentences. One computationally expensive solution is to train the language generation model for each possible word class. Instead, we use a single language generation model trained on input sentences with 66% of the input words randomly removed. We train this model once and then selectively drop out words during testing. This approach is effectively a de-noising auto-encoder, where noise takes the form of removed words. Using this trained auto-encoder we plan to search for a simple part-of-speech removal sentence compression scheme, so that when a fixed fraction of tokens are removed the reconstruction loss is minimised. This compression is inherently lossy, and the compressed text need not make grammatical sense making this task distinct from the human readable sentence compression task reviewed in Section 5.2.2.

Our search for the most important word classes starts with uniform random removal of all words down to the 33% level and thereby establishes a baseline. From there, each possible word class is given a rank; higher ranked word classes are always completely removed before lower ranked word classes; removal stops when only 33% of words remain. Words from classes of the same rank are chosen uniformly at random. For example, if the input sentence is *“the cat on the mat .”* and the removal order had nouns ranked 2 and all other parts of speech ranked 1, then nouns *“cat”* and *“mat”* would both be removed. Remaining words would be randomly removed until only 2 out of the 6 remain. Using this method we should see the lowest perplexity when the words are ordered from least important to most important.

Our forward selection approach tries to set each word type to the highest non-occupied rank or the lowest non-occupied rank. The selection which minimises the perplexity is then fixed and the search proceeds until all classes are ranked. The final ordering was **adjective, adverb, coordinating conjunction, particle, determiner, preposition or subordinate conjunction, verb, pronoun and noun**, with adjective judged the least useful and noun the most useful. Adjectives lack importance perhaps because they have only a local effect on a sentence and are often poorly detected by the CNN+RNN systems [Anderson et al., 2016; Vinyals et al., 2015b]. This ordering is in line with our term space construction rules presented in Section 6.2.1.

While our approach involves removing all occurrences of a particular POS class, this is clearly not the ideal scenario, since words within each class carry a variable amount of visual information. For example the adjectives, *“tall”*, *“red”*, and *“reflective”* all have a sense with a clear visual grounding, while the adjectives *“happy”*, *“honest”*, and *“angry”* have a less clear visual grounding. While previous work has looked at identifying visual vs non-visual attributes [Yanai and Barnard, 2005; Berg et al., 2010; Xie and He, 2013], to use these methods in practice would require ac-

curate word sense disambiguation. As we previously noted existing models seem to work poorly on image captions perhaps due to data mismatch. The range of attributes that the CNN can detect reliably is another constraint, for instance Anderson et al. [2016] calculate F-scores for different attribute classes produced by numerous state-of-the-art image captioning techniques (trained on MSCOCO). While the best methods could reliably detect colour attributes, all methods were relatively poor at identifying attributes in general, and especially poor at counting objects. As standard captioning methods continue to improve, developing a flexible representation for attributes will become more important for styled caption generation, but we leave this for future investigation.

6.2.3 Generating semantic sequences from images

We design a *term generator* network that maps an input image, denoted I , to an ordered sequence of semantic terms $\mathbf{x} = \{x_1, x_2, x_i, \dots, x_M\}, x_i \in \mathcal{V}^{term}$. The *term generator* is responsible for all semantic content; it must identify important concepts and provide a self consistent list of semantic terms. For example, term lists should not contain duplicate concepts “cow”, “cattle”, “livestock” unless they each need to be described separately, nor should they be missing important concepts. The following *language generator* assumes all semantic terms are correct and necessary for forming a caption.

Our *term generator* network is a CNN+RNN structure inspired by Show and Tell [Vinyals et al., 2015b], and illustrated in the lower left of Figure 6.2. The image features are extracted from the second last layer of the Inception-v3 [Szegedy et al., 2016] CNN trained on ImageNet [Russakovsky et al., 2015]. They then pass through a densely connected layer before being provided as input to an RNN with Gated Recurrent Unit (GRU) cells. The term list x is shorter than a full sentence, which speeds up training and alleviates the effect of forgetting long sequences.

At each time-step i , there are two inputs to the GRU cell. The first is the previous hidden state \mathbf{h}_{i-1} summarising the image I and term history x_1, \dots, x_{i-1} ; the second is the embedding vector \mathbf{E}_{x_i} of the current term. A fully connected layer with softmax non-linearity takes the output \mathbf{h}_i and produces a categorical distribution for the next term in the sequence x_{i+1} . Argmax decoding can be used to recover the entire term sequence from the conditional probabilities:

$$x_{i+1} = \underset{j \in \mathcal{V}^{term}}{\operatorname{argmax}} P(x_{i+1} = j | I, x_1 \dots x_i) \quad (6.1)$$

We set x_1 to be a beginning-of-sequence token and terminate when the sequence exceeds a maximum length or the end-of-sequence token is generated.

6.2.4 Generating styled descriptions

The *language generator*, shown in the upper right of Figure 6.2, maps from a list of semantic terms to a sentence with a specific style. For example, given the term list “man_NOUN”, “Competition_FrameNet”, “football_NOUN”, a suitable target can be “The handsome man played football like there was no tomorrow.”. Given the list of semantic terms \mathbf{x} , we generate an output caption $\mathbf{y} = \{y_1, y_2, y_t, \dots, y_L\}$, $y_t \in \mathcal{V}^{out}$ – where \mathcal{V}^{out} is the output word vocabulary. To do so, we learn an RNN sequence-to-sequence *language generator* network with attention over the input sequence, using styled text without corresponding paired images.

The encoder component for sequence \mathbf{x} consists of a Bidirectional RNN [Schuster and Paliwal, 1997b] with GRU cells and a learn-able term to vector embedding. The Bidirectional RNN is implemented as two independent RNNs running in opposite directions with shared term embeddings. Hidden outputs from the forward RNN $\mathbf{h}_{fwd,i}$ and the backward RNN $\mathbf{h}_{bak,i}$ are concatenated to form the hidden outputs of the encoder $\mathbf{h}_{enc,i} = [\mathbf{h}_{fwd,i}, \mathbf{h}_{bak,i}]$. The last of these hidden outputs is used to initialise the hidden state of the decoder $\mathbf{h}_{dec,0} = \mathbf{h}_{enc,M}$. The decoder itself is a unidirectional RNN (only a single forwards RNN) with GRU cells, learn-able word embeddings, an attention layer, and a softmax output layer.

The attention layer connects selectively weighted encoder hidden states directly to decoder cells, using weightings defined by a learnt similarity (Equations 6.2 & 6.3). This avoids compressing the entire sequence into a single fixed length vector, which improves performance in sequence-to-sequence modelling [Wu et al., 2016; Sutskever et al., 2014; Luong et al., 2015]. Attention vector $\mathbf{a}_t = (a_{t,1}, \dots, a_{t,i}, \dots, a_{t,M})$ quantifies the importance of the input term i to the current output time-step t . We compute the attention vector as a softmax over similarity \mathbf{v}_t with learnt weight matrix W^a , defined as:

$$\begin{aligned} v_{t,i} &= \mathbf{h}_{enc,i}^\top W^a \mathbf{h}_{dec,t} \\ a_{t,i} &= \exp(v_{t,i}) / \sum_{j=1}^M \exp(v_{t,j}) \end{aligned} \quad (6.2)$$

Using the attention vector, we compute a context vector that summarises the important hidden outputs of the encoder for the current decoder time step. The context vector at step t is defined as a weighted sum of the hidden outputs:

$$\mathbf{c}_t = \sum_{i=1}^M a_{t,i} \mathbf{h}_{enc,i} \quad (6.3)$$

To produce the output distribution, we concatenate the context vector \mathbf{c}_t with the hidden output of the decoder component $\mathbf{h}_{dec,t}$, and apply a fully connected layer

with softmax non-linearity:

$$\begin{aligned} \mathbf{h}_{out,t} &= W^{out}[\mathbf{c}_t, \mathbf{h}_{dec,t}] + \mathbf{b}^{out} \\ p(y_t = k|\mathbf{x}) &= \exp(h_{out,t,k}) / \sum_{j=1}^{|\mathcal{V}^{out}|} \exp(h_{out,t,j}) \end{aligned} \quad (6.4)$$

Here $|\mathcal{V}^{out}|$ denotes the output vocabulary size; $[\mathbf{c}_t, \mathbf{h}_{dec,t}]$ denotes vector concatenation; $W^{out}, \mathbf{b}^{out}$ are both learnt parameter of the output layer; and t is an index to the current element of the decoded sequence.

The sequence x is input to the *language generator* in reverse order because this is known to improve performance [Sutskever et al., 2014] by reducing the minimum number of steps between input and loss function – effectively reducing the depth. Masking is applied in both the encoder and decoder component to allow for variable length sequences during training. End-of-sequence tokens enable variable length outputs at generation time.

6.3 Learning with Unpaired Styled Texts

To learn SemStyle, we train *term generator* and *language generator* separately on different datasets. An existing image-caption dataset with factual descriptions is used for training the *term generator*, while a large set of styled sentences without aligned images in addition to factual captions are used for training the *language generator*. Our intermediate representation does not need to be learnt because it is defined by the steps in Section 6.2.1.

6.3.1 Training the term generator

We train the *term generator* network on an image caption dataset with factual descriptions, such as MSCOCO. The ground truth semantic sequence for each image is constructed from the corresponding ground truth descriptive captions by following the steps in Section 6.2.1.

For each image, the loss function is the mean categorical cross entropy over semantic terms in the sequence:

$$\mathcal{L} = -\frac{1}{M} \sum_{i=1}^M \log p(x_i = \hat{x}_i | I, \hat{x}_{i-1} \dots \hat{x}_1) \quad (6.5)$$

Here \hat{x} denotes ground truth terms. At training time the input terms $\hat{x}_{i-1} \dots \hat{x}_1$ are ground truth – this is the common teacher forcing technique [Williams and Zipser,

1989]. We found that schedule sampling [Bengio et al., 2015] – where sampled outputs are fed as inputs during training – did not improve performance, despite recent work on longer sequences achieving small gains [Vinyals et al., 2017].

6.3.2 Training the language generator

The *language generator* described in Section 6.2.4 takes a semantic term sequence \mathbf{x} as input and generates a sentence \mathbf{y} in the desired style. To create training data, we take a training sentence \mathbf{y} and map it to a semantic sequence \mathbf{x} according to the steps in Section 6.2.1. The loss function is categorical cross entropy.

We train the *language generator* with both styled and descriptive sentences. This produces a richer language model able to use descriptive terms that are infrequent in styled sentences. Training only requires text, making it adaptable to many different datasets.

Concatenating both datasets leads to two possible output styles; however, we wish to specify the style. Our solution is to provide a *target-style term* during training and testing. Specifically, our *language generator* network is trained on both the descriptive captions and the styled text with a *target-style term*, indicating provenance, appended to each input sequence. As our encoder is bidirectional we expect it is not sensitive to term placement at the beginning or end of the sequence, while a term at every time step would increase model complexity. This technique has previously been used in sequence-to-sequence models for many-to-many translation [Johnson et al., 2017]. In Section 6.5 we demonstrate that purely descriptive or styled captions can be generated from a single trained model by changing the *target-style term*.

6.4 Evaluation Setting

Both the *term generator* and *language generator* use separate 512 dimensional GRUs and term or word embedding vectors. Since the *term generator* uses a bidirectional RNN, each direction uses 256 dimensions. The maximum sequence length in both the *term generator* and the *language generator* is 22, including beginning and end of sequence tags. The vocabularies used by the *language generator* and the *term generator* are the most frequent terms or words in the training set, with separate vocabularies for the input words, semantic terms and output words. The *term generator* uses a vocabulary of 10000 terms, while the *language generator* uses a vocabulary of 20000 terms or words to account for a broader semantic scope: it must apply to both styled and descriptive sentences. With joint training the overlap between the *term generator* output terms and the *language generator* input is 8266 terms, without joint training

this falls to 6736 terms. Mismatches are replaced with the special “<unk>” token. The image input to the *term generator* is represented by 2048 dimensions from the second last layer of the Inception-v3 CNN [Szegedy et al., 2016]. We do not fine-tune Inception-v3, but doing so would likely improve semantic accuracy [Vinyals et al., 2015b; Gan et al., 2017b] equally for SemStyle and the baselines.

Learning uses mini-batch stochastic gradient descent method Adam [Kingma and Ba, 2015] with learning rate 0.001. We clip gradients to $[-5, 5]$ and apply dropout to image and sentence embeddings. The mini-batch size is 128 for both the *term generator* and the *language generator*. For the *language generator* each mini-batch is composed of 64 styled sentences and 64 image captions. To achieve this one-to-one ratio, we randomly down-sample the larger of the two datasets at the start of each epoch.

At test time both the *term generator* and the *language generator* use greedy decoding: the most likely word is chosen as input for the next time step. The code and trained models are released online.

6.4.1 Datasets

Descriptive image captions come from the MSCOCO dataset [Chen et al., 2015] of 82783 training images and 40504 validation images, with 5 descriptive captions each. It is common practice [Vinyals et al., 2015b] to merge a large portion of this validation set into the training set to improve captioning performance. We reserve 4000 images in the validation set as a test set, the rest we merged into training set. The resulting training set has 119287 images and 596435 captions.

The styled text consists of 1567 romance novels from bookcorpus [Zhu et al., 2015] – comprising 596MB of text and 9.3 million sentences. We filter out sentences with less than 10 characters, less than 4 words or more than 20 words. We further filter sentences not containing any of the 300 most frequent non stop-words from the MSCOCO dataset – leaving 2.5 million sentences that are more likely to be relevant for captioning images. Our stop-word list is from NLTK [Bird et al., 2009] and comparisons are on stemmed words (using porter stemmer). Stemming is used for efficiency rather than lemmatization as rough equivalence classes are sufficient for filtering the dataset. As a convenience for faster training time, we further down-sample to 578,717 sentences, with preference given to sentences containing the most frequent MSCOCO words. We remove all but the most basic punctuation (commas, full stops and apostrophes), convert to lower-case, tokenise and replace numbers with a special token.

The StyleNet [Gan et al., 2017a] test set was not released publicly at the time of

writing, so we could not use it for comparisons.

6.4.2 Baselines

Our evaluations included 6 state-of-the-art baselines.

CNN+RNN-coco is based on the Show+Tell model [Vinyals et al., 2015b] and trained on only the MSCOCO dataset. We use a GRU cell in place of an LSTM cell for a fairer comparison with our model. In fact, this baseline is just the *term generator* component of SemStyle trained to output full sentences. All hyper-parameter settings are the same as for *term generator*.

TermRetrieval uses the *term generator* to generate a list of terms – in this case the term vocabulary is words rather than lemmas with POS tags. These terms are used in an OR query of the Romance text corpus and scored with BM25 [Jones et al., 2000] using hyper-parameters $b = 0.75, k_1 = 1.2$. Our query engine is Whoosh², which includes a tokenizer, lower-case filter, and porter stem filter. This model cannot generate captions that are not part of the romance text corpus and the same set of terms always gives the same sentence.

StyleNet is our re-implementation of the method proposed by Gan et al. [2017a] – the original code was not released at the time of writing. We train it on the MSCOCO dataset and the Romantic text dataset. We followed Gan et al. [2017a] and made the following implementation choices to ensure a fair comparison with other baselines. We use Inception-v3 [Szegedy et al., 2016] features rather than ResNet152 [He et al., 2016] features, and a batch size of 128 for both datasets. When training on styled text, *StyleNet* requires random input noise from some unspecified distribution. We tried a few variations and found Gaussian noise with $\mu = 0$ and $\sigma = 0.01$ worked reasonably well. Gan et al. [2017a] suggested a training scheme where the training set alternates between descriptive and styled at the end of every epoch. We found this fails to converge, perhaps because our datasets are larger and more diverse compared with *FlickrStyle10k* used in the original implementation. *FlickrStyle10k* contains styled captions rather than sentences sampled from novels; however, it is not released at the time of writing. To ensure *StyleNet* converges on our dataset we alternate between the MSCOCO dataset and Romantic text dataset after every mini-batch – a strategy suggested by Luong et al. [2016] for multi-task sequence-to-sequence learning.

neural-storyteller consists of pre-trained models released by Kiros [2015] for generating styled image captions. This model first retrieves descriptive captions using k-nearest neighbours in a multi-modal space [Kiros et al., 2014]. This space is learnt by minimising projected distances between CNN image features (from VGG-19 [Si-

²<https://pypi.python.org/pypi/Whoosh/>

monyan and Zisserman, 2015]) and caption embeddings from a GRU. Retrieved captions are encoded into skip-thought vectors [Kiros et al., 2015], averaged, and then shifted by the mean skip-thought vector of the target style. The skip-thought vectors are trained on the entirety of bookcorpus [Zhu et al., 2015]. A skip-thought vector decoder learnt on the romance genre subset of bookcorpus (the same subset we have used for our models) generates the caption. *neural-storyteller* generates passages by repeatedly sampling the decoder. We use only the first sentence because long passages would be disadvantaged by the evaluation criteria. For more details of *neural-storyteller* see Section 2.8.

JointEmbedding, shown in Figure 6.3, uses a learnt multi-modal vector space as the intermediate representation. The image embedder is a projection of pre-trained Inception-v3 [Szegedy et al., 2016] features h_I , while the *sentence embedder* is a projection of the last hidden state of an RNN with GRU units h_{enc} . Formally the projections are:

$$\begin{aligned} v_I &= \tanh(W_I \cdot h_I) \\ v_s &= \tanh(W_s \cdot h_{enc}) \end{aligned}$$

Denoting the projections as, v_I for images and v_s for captions, and the learnt projection weights as W_I for images and W_s for captions. Agreement between image and caption embedding is defined as the cosine similarity:

$$g(v_I, v_s) = \frac{v_I \cdot v_s}{|v_I| |v_s|}$$

To construct the space we use a noise contrastive pair-wise ranking loss suggested by Kiros et al [Kiros et al., 2014]. Intuitively, this loss function encourages greater similarity between embeddings for paired image-captions than for un-paired images and captions.

$$\mathcal{L} = \max(0, m - g(v_I, v_s) + g(v_{I'}, v_s)) + \max(0, m - g(v_I, v_s) + g(v_I, v_{s'}))$$

Where s is the input caption paired with image I , while s' is a randomly sampled noise contrastive caption and I' the noise contrastive image. The margin m is fixed to 0.1 in our experiments.

The *sentence generator* is an RNN with GRU units that decodes from the joint vector space. The loss function is categorical cross entropy given in Equation 6.5.

Training is a two stage process. First, we define the joint space by learning the image embedder and the *sentence embedder* on MSCOCO caption-image pairs. From here on the parameters of image embedder and the *sentence embedder* are fixed. The *sentence generator* is learnt separately by embedding styled sentences from the romantic novel dataset with the *sentence embedder* into the multi-modal space and then

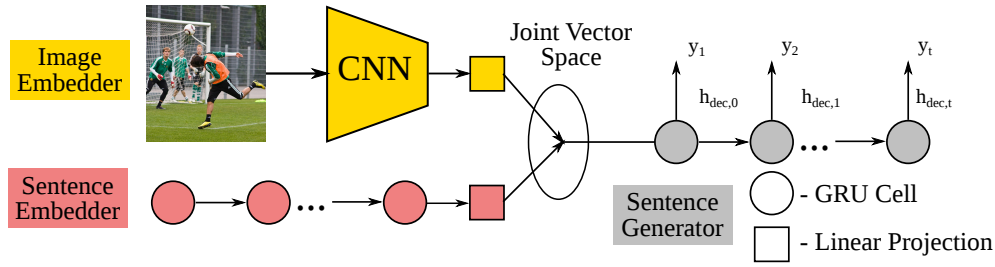


Figure 6.3: An overview of the *JointEmbedding* model. The two embedding components image embedder (in yellow) and sentence embedder (in red) are shown on the left while the sentence generator (in grey) is on the right.

attempting to recover the original sentence. This model has not been published previously, but is based on existing techniques for descriptive captioning [Kiros et al., 2014].

6.4.3 Model Variants

Our full model is denoted *SemStyle*. We use the following variants to assess several modelling choices in Section 6.5.4.

SemStyle-unordered is a variant of *SemStyle* with a randomised semantic term ordering. This model helps us to quantify the effect of ordering in the term space.

SemStyle-words is a variant where the semantic terms are raw words – they are not POS tagged, lemmatized or mapped to FrameNet frames.

SemStyle-lempos is a variant where the semantic terms are lemmatized and POS tagged, but verbs are not mapped to FrameNet frames. This helps us to quantify the degree to which verb abstraction affects the model performance.

SemStyle-romonly is *SemStyle* without joint training – the language generator was trained only on the romantic novel dataset. This model helps to quantify the effect of joint training.

SemStyle-cocoonly is the *SemStyle* model trained only on MSCOCO. The output of this model should be purely descriptive.

SemStyle-coco is the *SemStyle* model trained jointly on MSCOCO and the romance corpus. A MSCOCO *target-style term* used at test time indicates the output of this model should be purely descriptive.

6.4.4 Evaluation Metrics

Our evaluations use automatic metrics and human judgements for both content relevance and style. For human judgements we provide a detailed task description to ensure repeatability.

6.4.4.1 Automatic Metrics

Automatic content relevance metrics. Widely-used captioning metrics BLEU [Papineni et al., 2002], METEOR [Denkowski and Lavie, 2014] and CIDEr [Vedantam et al., 2015], which are based on n-gram overlap, have less relevance to the style generation. Using different wording is the goal of styled text generation, something these metrics punish heavily. The SPICE [Anderson et al., 2016] metric computes an f-score over semantic tuples extracted from MSCOCO reference sentences [Chen et al., 2015]. This is less dependent on exact n-gram overlap, and is strongly correlated with human judgements of descriptiveness. In our evaluation we include n-gram metrics for completeness, but refer mostly to the SPICE metric.

Automatic style metrics.

To the best of our knowledge, there are no well-recognised metrics for measuring image-captions conformance to a target style. We propose three metrics: two use a language model in the target style, the last is a high-accuracy style classifier. Similar metrics have been used for text only style generation, although the details are different [Xu et al., 2012]. Our first language model metric *LM* is a 4-gram model [Heafield et al., 2013] built on the styled text corpus – romance novels or COCO captions for this work. We report the average \log_2 perplexity per word (ie the average number of bits per word), with lower scores indicating stronger style. Our second language model metric *GRULM* is a GRU language model, also built on the style text corpus and reporting the average perplexity per word. Using both types of language model we hope to avoid unfairly biasing particular types of decoders. Moreover, these two language models help to assess the fluency of the generated text. The Classifier Fraction (CLF) metric is the fraction of generated captions classified as styled by a binary classifier. This classifier is logistic regression with 1,2-gram occurrence features trained on styled sentences and MSCOCO training captions. We use feature hashing, L2 regularization and grid-search cross-validation to choose hyper-parameters. The model’s cross-validation precision is 0.992 at a recall of 0.991. In Section 6.5.3 we calculate the correlation between each of these metrics and human style judgements. Models for all three evaluation metrics have been released.

Human evaluations of content and style. Automatic evaluation does not give a full picture of captioning systems performance [Chen et al., 2015]; human evaluation can help us to better understand their strengths and weaknesses with the end user in mind. We evaluate each image-caption pair with two crowd-sourced tasks on the CrowdFlower³ platform. The first measures how descriptive a caption is to an image on a four point scale – from unrelated (1) to clear and accurate (4). Figure 6.4 shows

³<https://www.crowdflower.com>

the instructions given to workers, while Figure 6.5 is an example question. The second task evaluates the degree of style transfer. We ask the evaluator to choose from three mutually exclusive options – that the caption: is likely to be part of a story related to the image (*story*), is from someone trying to describe the image to you (*desc*), or is completely unrelated to the image (*unrelated*). Figure 6.6 shows the instructions given to workers, while Figure 6.7 is an example question. Note that most sentences in a romance novel are not identifiably romantic once taken out of context. Being part of a story is a identifiable property for a single sentence. To judge style generation, Gan et al. [2017a] asked annotators to select the most attractive captions given the scenario of sharing on social media. We use our story question rather than the shareability question, as it more concisely captures the literary quality of the styled text. We separate the descriptiveness and story aspects of human evaluation, after pilot runs found that the answer to *descriptiveness* interferes with the judgement about being part of a story.

Using each method, we caption the same 300 random test images, and evaluated each with $n \geq 3$ workers, giving a total of at least 900 judgements per method. In most cases $n = 3$, typically being greater than 3 when a worker successfully challenges a hidden test question (as explained below). We aggregate these judgements by assigning each one a weight $1/n$, and calculating the weight normalised sum for each possible answer. In the case of descriptiveness judgements, a further summary statistic is calculated as the average descriptiveness score, with 1.0 being the least descriptive and 4.0 being the most descriptive.

To ensure reliable results we inject questions with known ground-truth, and require workers to maintain at least 70% accuracy on these questions. For our initial ground-truth, we manually labelled a small selection of questions judged to be clear exemplars. On a limited number of our ground-truth questions workers consistently made mistakes, so we revised the answers or removed these questions from the ground-truth. Because ground-truth is never re-used for the same worker, it acts as a limit on the number of tasks they can complete, so expanding the ground-truth is essential for large jobs. To expand our ground-truth, we followed the procedure suggested in the CrowdFlower documentation: manually review and then add questions that all three annotators agree upon.

6.5 Results

Table 6.2 summarises measurements of content relevance against factual (MSCOCO) captions. Table 6.3 and Figure 6.8 report automatic and human evaluations on caption style learned from romance novels.

Overview

Help us decide how well image captions relate to each image.

Steps

1. Examine the image.
2. Decide how well the caption relates to the image
3. Select the appropriate option

Rules & Tips

Rules:

- There are four possible choices for level of descriptiveness:
 - "Completely unrelated": the caption does not describe the image at all, nor does it have any of the right words for describing the image. This also includes captions that are not specific to any image.
 - "A few of the right words": the caption has some of the right words but they may be in the wrong order or used in a way that doesn't relate to the image
 - "Almost there, a few mistakes": has some of the main objects and/or actions, clearly relates to the image but has some mistakes such as: confusing male and female, using the wrong colors or adjectives, using the wrong action or missing some of the contents.
 - "A clear and accurate caption perhaps with extra non-visual information.": a caption related to this image that may have additional non-visual or contextual information. It doesn't have to describe everything only the main object/s and or actions.

Tips:

- "Completely unrelated" also refers to captions that would work with almost any image, (ie they are not specific)
- "A clear and accurate caption" does not need to use perfect grammar but should be understandable and clear.

Figure 6.4: A screen-shot of the instructions provided to workers when evaluating the relevance of a caption to an image.



I had a glass vase filled with flowers, using it to block out all of it.

How well does this caption relate to the image? (required)

- ☐ Completely unrelated
- ☐ A few of the right words
- ☐ Almost there, a few mistakes.
- ☐ A clear and accurate caption, perhaps with extra non-visual information.

Figure 6.5: A screen-shot of a single question put to workers during the relevance evaluation task.

Overview

Help us decide which sentences related to images could come from a story about the image or are more likely to be only a description of the image.

Steps

1. Examine the image.
2. Decide if the sentence is related to the image
3. If it is, then decide if it came from a story or from someone trying to describe the image contents to you.

Rules & Tips

Rules:

- The sentence **does not** have to describe the image perfectly, but should relate to the image
- The sentence **does not** have to use perfect grammar
- If the sentence is **completely unrelated** to the image then select "The sentence is completely unrelated to the image."

Tips:

- Stories may use the first person eg "I went to the store", while a description would not.
- Stories often use more colorful and emotive language eg "The tranquil lake shimmered in the dawn light."
- Stories might refer to state of mind eg "I thought about eating the donut."
- Descriptions tend to be in present tense, relatively short and direct eg "A dog on some grass", "The pizza is sitting on a table"

Figure 6.6: A screen-shot of the instructions provided to workers when evaluating the conformance of a caption to the desired style.



I had a glass vase filled with flowers, using it to block out all of it.

Is this sentence from a story about the image or from someone trying to describe the image to you? (required)

- ☐ Story
- ☐ Description
- ☐ The sentence is completely unrelated to the image.

Figure 6.7: A screen-shot of a single question asked of workers in the style evaluation task.

<i>Model</i>	<i>BLEU-1</i>	<i>BLEU-4</i>	<i>METEOR</i>	<i>CIDEr</i>	<i>SPICE</i>	<i>CLF</i>	<i>LM</i>	<i>GRULM</i>
CNN+RNN-coco	0.667	0.238	0.224	0.772	0.154	0.001	6.591	6.270
StyleNet-coco	0.643	0.212	0.205	0.664	0.135	0.0	6.349	5.977
SemStyle-cocoonly	0.651	0.235	0.218	0.764	0.159	0.002	6.876	6.507
SemStyle-coco	0.653	0.238	0.219	0.769	0.157	0.003	6.905	6.691

Table 6.2: Evaluating caption descriptiveness on MSCOCO dataset. For metrics see Sec. 6.4.4, for approaches see Sec. 6.4.2.

<i>Model</i>	<i>BLEU-1</i>	<i>BLEU-4</i>	<i>METEOR</i>	<i>CIDEr</i>	<i>SPICE</i>	<i>CLF</i>	<i>LM</i>	<i>GRULM</i>
StyleNet	0.272	0.099	0.064	0.009	0.010	0.415	7.487	6.830
TermRetrieval	0.322	0.037	0.120	0.213	0.088	0.945	3.758	4.438
neural-storyteller	0.265	0.015	0.107	0.089	0.057	0.983	5.349	5.342
JointEmbedding	0.237	0.013	0.086	0.082	0.046	0.99	3.978	3.790
SemStyle-unordered	0.446	0.093	0.166	0.400	0.134	0.501	5.560	5.201
SemStyle-words	0.531	0.137	0.191	0.553	0.146	0.407	5.208	5.096
SemStyle-lempos	0.483	0.099	0.180	0.455	0.148	0.533	5.240	5.090
SemStyle-romonly	0.389	0.057	0.156	0.297	0.138	0.770	4.853	4.699
SemStyle	0.454	0.093	0.173	0.403	0.144	0.589	4.937	4.759

Table 6.3: Evaluating styled captions with automated metrics. For *SPICE* and *CLF* larger is better, for *LM* & *GRULM* smaller is better. For metrics and baselines see Sec. 6.4.4 and Sec. 6.4.2.

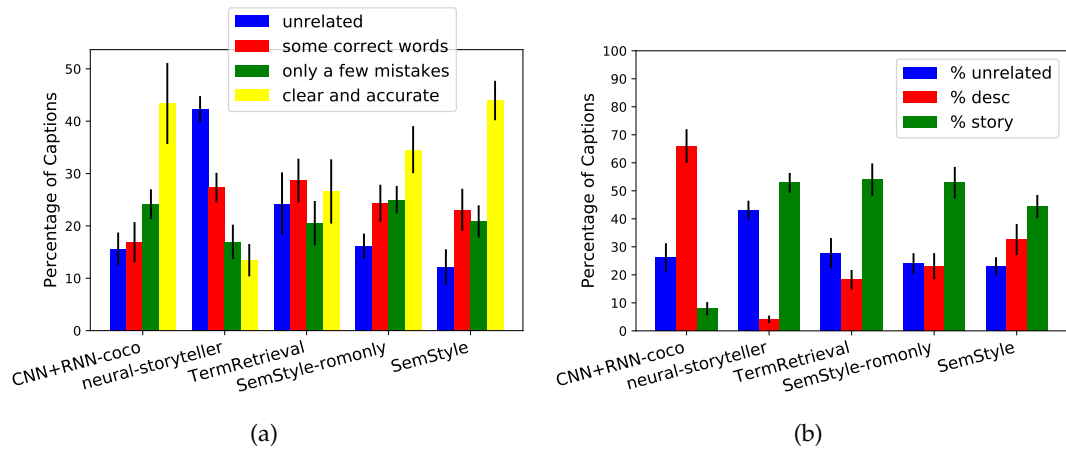


Figure 6.8: Human evaluations for SemStyle and selected baselines, with error bars showing 0.95 confidence intervals over 10 random splits. (a) descriptiveness measured on a four point scale, reported as percentage of generated captions at each level. (b) style conformity as a percentage of captions: unrelated to the image content, a basic description of the image, or part of a story relating to the image.

6.5.1 Evaluating Relevance

SemStyle-coco has the descriptive *target-style term* appended to the input, conditioning it to generate descriptive captions. It achieves semantic relevance scores comparable to *CNN+RNN-coco*, with a SPICE of 0.157 vs 0.154, and BLEU-4 of 0.238 for both. As the *term generator* is based on the *CNN+RNN-coco* the similar SPICE scores demonstrate that our semantic term representation is a competitive way to distil image semantics, and that the *term generator* and *language generator* constitute an effective vision-to-language pipeline.

When *SemStyle* is conditioned to generate styled sentences using the romance *target-style term*, it still retains a high degree of relevance as judged by the SPICE score of 0.144 in Table 6.3. This is a higher SPICE score than achieved by other methods of styled caption generation, with the next best method *TermRetrieval* achieving a SPICE of only 0.088. It is also in-line with other *SemStyle* variants; however, the *SemStyle-words* and *SemStyle-lempos* variants which keep intermediate terms closer to their surface form have slightly higher SPICE. As expected, the n-gram overlap metrics BLEU, METEOR, and CIDEr show a large disparity between styled and descriptive captions because of a sensitivity to the surface form.

6.5.2 Evaluating Style

SemStyle succeeds in generating styled captions in 58.9% of cases, as judged by CLF, and receives a SPICE score of 0.144. The baselines *TermRetrieval*, *neural-storyteller* and *JointEmbedding* have significantly higher CLF scores, but much lower SPICE scores. *TermRetrieval* produces weakly descriptive sentences (SPICE of 0.088) because it is limited to reproducing the exact text of the styled dataset, which yields lower recall for image semantics. Both *neural-storyteller* (SPICE 0.057) and *JointEmbedding* (SPICE 0.046) decode from a single embedding vector, allowing less control over semantics than *SemStyle*. This leads to weaker caption relevance. *StyleNet-coco* produces factual sentences with comparable BLEU and SPICE scores. However, *StyleNet* produces styled sentences less frequently (CLF 41.5%) and with significantly lower semantic relevance – SPICE of 0.010 compared to 0.144 for *SemStyle*. We observe that the original *StyleNet* dataset [Gan et al., 2017a] mostly consists of factual captions rewritten by adding or editing a few words. The romance novels in the book corpus, on the other hand, have very different linguistic patterns to COCO captions. We posit that the factored input weights in *StyleNet* work well for small edits, but have difficulty capturing richer and more drastic changes. For *SemStyle*, the semantic term space and a separate *language generator* make it amenable to larger stylistic changes.

Language model perplexity scores, LM and GRULM, generally agree with CLF



Figure 6.9: Example results, including styled (**Story**) output from *SemStyle* and descriptive (**Desc**) output from *SemStyle-coco*. Four success cases are on the left (a,b,c,d), and two failures on the right (e,f).

scores, such that a high CLF is commonly paired with a low perplexity. The result for *neural-storyteller* is a notable exception, with high CLF and perplexity relative to other methods. Inspecting the generated sentences suggests this could be the result of poor sentence construction and grammar usage, to which the language model is sensitive. We also find that the perplexity of the 4-gram language model (LM) mostly agrees with the GRU language model (GRULM). One case where LM and GRULM are markedly different is for the *TermRetrieval* method, having LM of 3.758 and GRULM of 4.438. *TermRetrieval* retrieves captions rather than generating statistically likely captions: a process which tends to restrict vocabulary usage (for example see the Unique Verbs in Table 6.7). The more diverse word usage appears to be handled better by the n-gram based LM rather than the GRULM.

SemStyle can reliably generate multiple output styles without retraining. *SemStyle-coco* in Table 6.2 and *SemStyle* in Table 6.3 share the same parameters but have a different *target-style term* added at test time. This leads to purely descriptive text from *SemStyle-coco*, and romance style text from *SemStyle*. Judged by the CLF metric, *SemStyle-coco* produces romance styled text 0.3% of the time, while *SemStyle* produces romance styled text 58.9% of the time.

6.5.3 Human Evaluations

The crowd-sourced experiments are summarised in Figure 6.8. Figure 6.8(a) shows image-caption relevance judged on a scale of 1 (unrelated) to 4 (clear and accurate). StyleNet was not included in the human evaluations since it scored significantly worse than others in the automatic metrics, especially SPICE and LM. *SemStyle* has a mean relevance of 2.97, while *CNN+RNN-coco* has 2.95. In addition, only 12.2% of *SemStyle* captions are judged as *unrelated*, the lowest among all approaches. *SemStyle* produces *clear and accurate* captions 43.8% of the time, while *CNN+RNN-coco* produces them 43.4% of the time – both of these scores are significantly higher than other approaches, see Table 6.5 for significance tests. As the CNN+RNN architecture is the basis of the *term generator*, this indicates our semantic term mapping and separate styled *language generator* do not reduce the relevance of the generated captions. *TermRetrieval* has mean relevance 2.50, and *neural-storyteller* 2.02 – both significantly lower than *SemStyle*, see Table 6.5. *neural-storyteller* generates a large fraction of completely unrelated captions (42.3%) while *TermRetrieval* avoids doing so (24.4%). *SemStyle-romonly* produces significantly fewer *clear and accurate* captions than *SemStyle* (34.7% vs 43.8%), which demonstrates improved caption relevance when both training datasets are combined and the *target-style term* used.

Figure 6.8(b) summarises crowd-worker choices from the options *story-like*, *descriptive*, or *unrelated*. The two *SemStyle* variants have the lowest (< 25%) fraction of captions that are judged *unrelated*. *SemStyle* generates *story-like* captions 41.9% of the time, which is far more frequently than the *CNN+RNN-coco* trained on MSCOCO at 6.2% – significance tests for *story-like* judgements are provided in Table 6.4. *neural-storyteller* produces captions that are judged as *story-like* 52.6% of the time, but at the expense of 44.2% completely unrelated captions. *TermRetrieval* produces captions that are *story-like* 55.5% of the time and unrelated only 26.0% of the time; however, as shown in Figure 6.8(a), they are consistently rated as having a low degree of relevance to images.

We follow the methodology of Anderson et al. [2016] to measure correlation between human judgements and captioning metrics; however, our metrics measure style conformity rather than descriptiveness. Specifically we calculate Kendall’s τ correlation co-efficient between the human *story-like* judgements and the automatic metrics: CLF, LM, and GRULM. Human judgements are averaged across the three annotators before correlation. Anderson et al. [2016] use captions sourced from a selection of high performing captioning methods – those that do well in the MSCOCO captioning competition [Chen et al., 2015]. Since no such competition exists for styled captions, we instead use captions generated by the five methods listed in Figure 6.8

	CNN+RNN-coco	neural-storyteller	TermRetrieval	SemStyle-romonly
<i>CNN+RNN-coco</i>	-	-	-	-
<i>neural-storyteller</i>	5.6e-09*	-	-	-
<i>TermRetrieval</i>	1.2e-08*	0.88	-	-
<i>SemStyle-romonly</i>	2.1e-12*	0.18	0.13	-
<i>SemStyle</i>	1.4e-06*	0.27	0.34	0.014

Table 6.4: χ^2 tests on method pairs for **human story judgements**. We combine counts for “unrelated” with “purely descriptive”, while “story” is kept as its own class. Those marked with a * indicate rejection of the null hypothesis (H0: the two methods give the same multinomial distribution of scores) at p-value of 0.005 – this is p-value of 0.05 with bonferroni correction of 10 to account for multiple tests.

	CNN+RNN-coco	neural-storyteller	TermRetrieval	SemStyle-romonly
<i>CNN+RNN-coco</i>	-	-	-	-
<i>neural-storyteller</i>	1e-56*	-	-	-
<i>TermRetrieval</i>	4.1e-18*	9.3e-14*	-	-
<i>SemStyle-romonly</i>	0.00032*	2.3e-35*	3.4e-07*	-
<i>SemStyle</i>	0.18	2.1e-48*	1.7e-13*	0.023

Table 6.5: χ^2 tests on method pairs for **human descriptiveness judgements**. We combine counts for “clear and accurate” with “only a few mistakes”, and “some correct words” with “unrelated”. Those marked with a * indicate rejection of the null hypothesis (H0: the two methods give the same multinomial distribution of scores) at p-value of 0.005 – this is p-value of 0.05 with bonferroni correction of 10 to account for multiple tests.

– all those for which we have human judgements. CLF has a correlation of 0.434 with human judgements, which is significantly more than the correlation of 0.150 for the LM metric and 0.091 for the GRULM metric. This suggests discriminative methods such as CLF can be a fairly effective proxy for human judgements of image caption style. However, we advise caution since the correlations may be affected by the methods used to generate the pool of captions.

6.5.4 Evaluating Modelling Choices

The last 5 rows of Table 6.3 highlight trade-offs among variants of *SemStyle*. Randomly ordering the semantic terms during training and testing – as in *SemStyle-unordered* – leads to captions with lower semantic relevance, shown by a SPICE of 0.134 compared to 0.144 for the full model. They also conform less to the target style with a CLF of 0.501 compared to 0.589.

Using a raw word term space *SemStyle-words* (without FrameNet, lemmatization

or POS tags) gives similar semantic relevance, SPICE of 0.146 to the full models 0.144, but less styling with CLF at 0.407. Using verb lemmas rather than FrameNet terms as in *SemStyle-lempos* has a similar effect, with a slight increase in SPICE to 0.148 and a decrease in style to a CLF of 0.533. This clearly demonstrates the three components FrameNet, lemmatization and POS tags all contribute to remove style from the intermediate representation, and thus lead to output in the target style.

Learning from both datasets improves caption relevance. If we only train on the romantic novel corpus as in *SemStyle-romonly*, we find strong conformity to the target style (CLF 0.770) but less semantic relevance, SPICE 0.138. Without the joint training, some semantics terms from the MSCOCO dataset are never seen by the language generator at training time – meaning their semantic content is inaccessible at test time. Our joint training approach avoids these issues and allows style selection at test time.

6.5.5 Example Captions

Figure 6.9 shows four success cases on the left (a,b,c,d) outlined in green, and two failure cases (e,f) on the right outlined in red. The success cases are *story-like*, such as (c) “*The woman stepped underneath her umbrella and walked in the rain.*” rather than “*A woman walking with an umbrella in the rain.*”. Case (c) also demonstrates the use of past tense and definite articles, which is a property of romance dataset but not MSCOCO – see Section 6.5.8. *SemStyle* captions also tend to use more interesting verbs due to FrameNet verb abstraction. For example (a) “*He pulled out a horse carriage and **charged** down the street.*”. Usage of first person perspective is demonstrated in (e,f). The failures are caused by: the *term generator* incorrectly identifying cows in the image (e), and the *language generator* using the word “*juicer*” in a way that is grammatically correct, but contradicts common-sense (f). In caption (f) it is also apparent that there is additional information about the actor or speaker “*I’ll be in*” that has been invented. This occurs because, during training on romance novels (where this utterance is frequent), none of the words in “*I’ll be in*” are included in the list of input semantic term, thus the *language generator* learns to generate these words without specific conditioning. The definition of the term space has an effect on what content is preserved from the images, and also what can be invented by the *language generator*.

6.5.6 Coverage of Semantic Terms

The *language generator* is trained to generate text that includes all semantic input terms. This is not implemented as a specific constraint but as a property of the train-

<i>Model</i>	<i>BLEU-1</i>	<i>ROUGE-1</i>
CNN+RNN-coco	0.561	0.517
StyleNet-coco	0.506	0.468
SemStyle-cocoonly	0.636	0.531
SemStyle-coco	0.631	0.532
StyleNet	0.027	0.028
TermRetrieval	0.505	0.336
neural-storyteller	0.234	0.225
JointEmbedding	0.340	0.177
SemStyle-unordered	0.597	0.501
SemStyle-words	0.611	0.517
SemStyle-lempos	0.593	0.504
SemStyle-romonly	0.624	0.511
SemStyle	0.626	0.517

Table 6.6: Precision (BLEU-1) and recall (ROUGE-1) in our semantic term space.

ing data: by construction, all input terms are reflected in the output. We can evaluate how much use the trained *language generator* makes of this term space by matching the input terms (the output of the *term generator*) to the POS tagged, lemmatized, and SEMAFOR parsed output captions. For *SemStyle*, all non-FrameNet input terms were used, and had the correct POS tag, 94% of the time, while all FrameNet terms were used 96% of the time. This strong relationship between the semantic input terms and the output sentence helps to ensure caption relevance.

We can extend this analysis to other methods by matching semantic terms in the output sentence with semantic terms in the caption ground truth. Unlike the previous evaluation, the results will depend on the efficacy of the visual concept detection pipeline (eg the *term generator* for *SemStyle*). As we are evaluating with respect to our term space, we expect a bias that favours models also using our terms space. The primary purpose of this analysis is therefore to confirm *SemStyle* accurately produces captions with term representations similar to the ground truth. Precision is reported as BLEU-1 without length penalty on terms, while recall is reported as ROUGE-1 on terms. Both BLEU-1 and ROUGE-1 are multi-reference metrics, allowing us to measure precision and recall against the 5 ground truth captions. Results in Table 6.6 show that the four variants of *SemStyle* (*SemStyle-cocoonly*, *SemStyle-coco*, *SemStyle-romonly*, *SemStyle*) that use our semantic term space perform better than methods that do not use our term space. This demonstrates *SemStyle* focuses on accurate reproduction of the semantic term space. The best performing models are *SemStyle-cocoonly* with the largest BLEU-1 and *SemStyle-coco* with the largest ROUGE-1 – though both models score highly in BLEU-1 and ROUGE-1. This is in line with

the other automatic metrics shown in Table 6.2, although these metrics also show *CNN+RNN-coco* is competitive. Of the baselines the best performing is *TermRetrieval*, which retrieves romance sentences using query words from a *term generator* (trained only on raw words in this case).

6.5.7 Diversity

Generating a diverse set of captions, is an important goal as it keeps captions interesting and can be seen as an aspect of model flexibility. Using exact string comparisons between captions we find *SemStyle* generates a relatively diverse range of captions. From the 4000 test images *SemStyle* generates 2308 unique captions. For reference, the descriptive baseline *CNN+RNN-coco* generates 2704 unique captions on the same set. The diversity of *SemStyle* is limited by the diversity of the *term generator*: an identical term list leads to an identical output sentence. In comparison, models which use vector intermediate representations generate more unique captions, with: *JointEmbedding* at 3711 unique captions, and *neural-storyteller* at 3975 unique captions. However, as noted previously, these models produce captions that are significantly less relevant to the image. If additional output diversity is required from *SemStyle*, sampling rather than argmax decoding may be used in either the *term generator* or the *language generator*—possibly at the expense of relevance and grammar.

6.5.8 Exploring the Generated Style

The style of the text is difficult to define in its entirety, but we can look at a few easily identifiable style attributes to better understand the style introduced into the captions. As a basis for comparison we randomly sample 4000 captions or sentences from the MSCOCO and romance dataset. We then generate captions for 4000 images using *SemStyle* and *CNN+RNN-coco*. On these four datasets we count: the percentage of sentences with past or present tense root verbs (to identify the tense used in the captions), the percentage of sentences with first person pronouns (to identify sentences using first person perspective), the number of unique verbs used in the 4000 samples (to identify verb diversity). The results are summarised in Table 6.7. Part-Of-Speech (POS) tags and syntactic dependency relationships are obtained automatically with the spaCy⁴ library. For counting purposes, past tense verbs are those tagged with Penn Treebank tags VBD, while present tense verbs are those tagged with VBZ or VBP. For VBN and VBG we adopt the tense of the auxiliary if one exists, failing that the sentence is marked as neither past nor present. To avoid cases where multiple tenses are present in a single sentence we count only the tense of the root

⁴<https://github.com/explosion/spaCy/tree/v1.9.0>

	Sentences with Present Tense Root Verbs	Sentences with Past Tense Root Verbs	Sentences with First Person Pronouns	Unique Verbs
<i>MSCOCO ground-truth</i>	32.2%	0.9%	0.2%	497
<i>romance ground-truth</i>	20.2%	59.3%	31.2%	1286
<i>CNN+RNN-coco</i>	34.1%	0.2%	0.0%	181
<i>SemStyle</i>	8.58%	64.6%	24.4%	348

Table 6.7: Style attribute statistics based on 4000 random ground-truth sentences for MSCOCO and romance styles and 4000 test captions generated by the descriptive only model (*CNN+RNN-coco*) and our *SemStyle* model. We measure the percentage of sentences or captions with present tense root verbs, past tense root verbs, and first person pronouns. We also count the number of unique verbs used in the sampled sentences.

verb identified by the syntactic dependency parse. Sentences without a root verb are marked as neither past nor present.

As shown in Table 6.7, captions generated by *SemStyle* have a past tense verb as the root verb in 64.6% of sentences, which is close to the romance ground-truth level of 59.3% and far greater than the descriptive method (*CNN+RNN-coco*) at 0.2%. This corresponds to a reduction in present tense verb usage; however, the ground-truth romance sentences include a greater fraction of sentences with present tense root verbs. *SemStyle* includes first person pronouns in 24.4% of captions, compared to 0.0% for *CNN+RNN-coco*. The *romance ground-truth* has personal pronouns in 31.2% of sentences, which is higher than *SemStyle* – we expect that describing images limits the applicability of first person pronouns. *SemStyle* has an effective verb vocabulary almost twice as large (92.3% larger) as *CNN+RNN-coco*, which suggests more interesting verb usage. However, both *SemStyle* and *CNN+RNN-coco* have lower verb diversity than either ground-truth dataset. In part this can be explained by argmax decoding tending to generate more common words. Additionally, we expect many of the verbs used in the *romance ground-truth* cannot be readily applied to image captioning. Overall, we find that the *SemStyle* model reflects the ground-truth romance style, generating more captions in past tense, first person, and with greater verb diversity.

To further explore the differences between styles we include Table 6.8, presenting the most common lemmas for each dataset stratified by POS tag. The most common nouns generated by *SemStyle* have a greater overlap with the *MSCOCO ground-truth* than the *romance ground-truth*. This is the desired behaviour since nouns are a key

Word Source	Most Common Lemmas
<i>MSCOCO ground-truth</i>	
NOUN	man(3.7%), people(1.9%), woman(1.8%), street(1.5%), table(1.4%)
VERB	be(20.0%), sit(9.3%), stand(6.4%), hold(4.4%), ride(3.1%)
ADJ	white(6.8%), large(5.4%), black(4.1%), young(4.0%), red(3.8%)
DET	a(81.8%), the(14.9%), some(1.7%), each(0.6%), this(0.4%)
<i>romance ground-truth</i>	
NOUN	man(2.7%), hand(1.5%), eye(1.4%), woman(1.3%), room(1.2%)
VERB	be(15.5%), have(4.6%), do(2.5%), would(2.4%), can(1.9%)
ADJ	small(2.3%), other(2.0%), little(2.0%), black(2.0%), white(1.9%)
DET	the(60.5%), a(26.5%), that(3.2%), this(2.8%), no(1.3)%
<i>CNN+RNN-coco</i>	
NOUN	man(6.9%), group(3.0%), people(2.6%), table(2.6%), field(2.3%)
VERB	be(29.4%), sit(15.4%), stand(10.2%), hold(5.6%), ride(4.6%)
ADJ	large(15.0%), white(10.9%), green(4.7%), blue(4.5%), next(4.5%)
DET	a(91.9%), the(7.7%), each(0.2%), some(0.1%), an(0.1%)
<i>SemStyle</i>	
NOUN	man(5.5%), table(2.8%), street(2.7%), woman(2.6%), who(2.4%)
VERB	be(24.5%), sit(10.3%), stand(4.8%), have(3.6%), hold(3.2%)
ADJ	sure(14.7%), little(9.4%), hot(5.6%), single(4.7%), white(3.9%)
DET	the(68.6%), a(30.8%), no(0.2%), any(0.2%), an(0.1%)

Table 6.8: The most common words per part-of-speech category in the two ground-truth datasets and in the sentences generated by the descriptive model (*CNN+RNN-coco*) and *SemStyle*. For each word we display the relative frequency of that word in the POS category – represented as a percentage.

component of image semantics and so nouns generated by the *term generator* should be included in the output sentence. The most common verbs generated by *SemStyle* are also similar to the *MSCOCO ground-truth*; we expect this is a result of a similar set of common verbs in both ground-truth datasets. The use of determiners in *SemStyle* more closely matches the *romance ground-truth*, in particular the frequent use of the definite article “*the*” rather than the indefinite “*a*”. The most common adjectives in all word sources typically relate to colour and size, and vary little across the different sources.

6.6 Summary

I propose *SemStyle*, a method to learn visually grounded style generation from texts without paired images. I develop a novel semantic term representation to disentangle content and style in descriptions. Since this term representation captures content from either captions or styled sentences, we are able to learn a mapping from an image to a sequence of semantic terms that preserves visual content, and a decoding

scheme that generates a styled description. Key to this model is the separation of concerns: the *term generator* focuses on visual concept detection and content planning, while the *language generator* focuses on style and language.

One significant obstacle encountered during this work was the domain difference between image captions and sentences from romance novels. Image captions are rich in visual concepts, while sentences from romance novels are not visually grounded and often refer to abstract concepts. However, when trying to generate captions in a romantic style, frequent use of visual concepts is necessary. In effect, the target captions should not conform exactly to the romantic text or the descriptive captions, but instead be a coherent mixture of the two. Without aligned images and styled text it is difficult to define how this mixture should be constructed. In the case of *SemStyle* the mixture is primarily defined by the shared term space, although the *language generator* has veto power when the input terms cannot be easily formed into a styled sentence. A better result could possibly be achieved by learning to mix the descriptive captions with the styled text using a small number of aligned images and styled captions.

Other future work includes learning from a richer set of styles, and developing a recognised set of automated and subjective metrics for styled captions.

Conclusion

In this chapter I summarise the contributions made by this thesis with regard to the three core challenges identified in Chapter 1. I suggest some interesting directions for future work on styled image caption generation and related problems.

7.1 Summary

This thesis has presented novel methods for generating styled natural language descriptions for images as well as novel approaches to related tasks. Chapter 3 introduced a novel approach to selecting names for visual concepts using context. This approach shows promise for choosing names that match a particular context or style. Moving from words to full sentences, Chapter 4 presents SentiCap: the first published system capable of automatically generating image captions in distinct styles. SentiCap generates both positive and negative sentiment captions, while requiring only a small set of styled training captions. Chapter 5 then specifically considered the language generation sub-task, introducing the S4 model with novel ideas for sentence simplification. This informed the development of linguistic components of a new styled caption generation approach called SemStyle, which was presented in Chapter 6. Unlike previous work, SemStyle generates visually relevant captions in a style defined by a large text corpus, with no styled captions required for training. Judged by crowd-sourced and automatic metrics, SemStyle captions have a recognisable style component and are descriptive of images.

At the start of this thesis I broke down styled image captioning into three core components: style and content representation, generative models for styled captions, and methods for overcoming data scarcity. Here I briefly summarise the contributions of this thesis towards these goals.

Representing style and content is a problem specific task. Throughout this thesis I tackle the problem in a number of different ways. Chapter 3 considers naming choice: style is encoded in the choice of names, while content is defined by the object

detectors. I show that this separation of content and style allows more natural names to be chosen; even allowing visual context to play a role. In Chapter 4 I consider positive and negative sentiment as stylistic variations, and design a data collection task for collecting captions aligned with these sentiment dimensions. A novel neural network model is then trained with this dataset, encapsulating the properties of the style within the parameters. The switching structure of the network allows for separate style and descriptive components. Chapter 6 introduces a novel term space that effectively captures the image content required for captioning. Compared to a vector representation of image content (CNN features), this term space leads to no significant loss in caption relevance as judged by both automatic and human evaluations. Moreover, the term space demonstrates a degree of style invariance that allows it to be used as an intermediate space for styled text generation, where the style is encoded in the parameters of the language generator.

Seamlessly incorporating both content and style attributes is pivotal to designing generative models for styled image captions. To achieve this goal I develop several conditional neural network language model variants that generate styled text when conditioned on semantic content. SentiCap (Chapter 4) balances two conditional language models with a novel switching component and loss function, allowing a trade-off between style and content to be learnt from data. It is also the first published model for explicitly generating image captions in a distinct style. S4 (Chapter 5) is a sentence simplification model with a number of novel adaptations to enhance parameter usage efficiency by encouraging input-output word copying. By learning when to copy and when to make changes I trade-off correct semantics (from the original sentence) with simplification and the chance of a semantic error. SemStyle (Chapter 6) generates captions from an ordered list of semantic terms; it is trained to use all semantic terms to generate the caption. At its core, SemStyle consists of a term generator and a language generator joined by these shared semantic terms. The term generator is responsible for detecting and choosing concepts to go into the caption, while the language generator realises these concepts in an appropriately styled sentence. I show, through extensive evaluations, that SemStyle frequently generates captions that both relate to the image and express the target style.

Data for styled image captioning is typically scarce because of the difficulty obtaining image-captions in each possible style. I took a multi-faceted approach to data scarcity, developing techniques for: using noisy image captions from social media, crowd-sourced data collection, fine-tuning existing models, and exploiting word embeddings. In Chapter 3 I describe how to name concepts with the help of image captions extracted from social media. Noise reduction was achieved by matching visual detections to captions via external resources, such as WordNet and ITIS, that

define semantic categories. To develop SentiCap, in Chapter 4, I tackle data scarcity by developing a crowd-sourcing method that guides annotators in the editing of factual captions to meet style objectives. SentiCap also fine-tunes descriptive only models to reduce the required volume of new data. S4, in Chapter 5, uses a novel loss function to exploit common similarities between input and output sentences, which reduces data requirements; however, pre-trained word vectors for uncommon words also play a role. SemStyle from Chapter 6 is trained without aligned image styled caption pairs; in many cases this vastly reduces the data collection effort required to generate styled captions.

Styled image captioning and, more generally, styled text generation with semantic content control is still in its infancy. As techniques for generating semantically relevant text advance, we are likely to see a shift of focus towards generating styled sentences. Such styled generation could personalise digital content, improving communication and easing information access. Just as current search engines return personalised results with high individual relevance, styled text generation could be used to adjust text to make it more accessible on an individual level. To realise this long-term goal, a significant number of challenges must be overcome. In the next section I briefly outline some of these challenges and make specific suggestions for future work.

7.2 Future Work

Context and visually grounded naming. In Chapter 3 I demonstrated that visual context is an important factor in object naming. There are many other forms of context that are not visual, for example: geographic region, whether the image is being tagged or captioned, and specialised knowledge. Exploring the effect of these forms of context on naming is an important direction for future work. While previous works have looked at the relationship between contextual factors and words used in posts [Pavalanathan and Eisenstein, 2015; Shoemark et al., 2017b] they did so without visual grounding. By using visual grounding, we can uncover contextual naming changes rather than word usage changes.

Captions with fine-grained emotion. The presented version of SentiCap model is limited to positive or negative sentiment, although using a richer set of emotions could be possible. There has been some work on mapping adjective noun pairs to emotions such as joy, anger, sadness and trust. These relationships could be used in much the same way as the sentiment vocabularies used in the SentiCap data collection process. However, the large number of different emotions would introduce an even more challenging data sparsity problem. This compounds even further when mixing different attributes (eg to convey both joy and trust). Some recent

techniques [Ficler and Goldberg, 2017; Oraby et al., 2017] (also see Section 2.6) train unified models that are able to mix attributes, even in ways not seen during training. To the best of our knowledge, these approaches have not been applied to image captioning, nor do they enforce the degree of semantic control required for captioning.

Sequence-to-Sequence with external dictionaries. The S4 model for simplifying sentences was not able to learn a broad range of word substitutions, primarily because of the limited aligned data and large vocabulary. As high quality sentence aligned data is not available in vast quantities, it would be particularly helpful to exploit word substitution dictionaries from other tasks. Some authors have already attempted this [Napoles et al., 2016; Xu et al., 2016], with the machine translation defined paraphrase database PPDB [Pavlick and Callison-burch, 2013] being a key component. Integrating such external databases into neural network language models is still an open problem. Progress in this area has been made by models for out-of-domain image captioning [Tran et al., 2016; Anderson et al., 2017; Anne Hendricks et al., 2016]; however, applications to text simplification needs further exploration.

Metrics for caption style. Evaluating styled image captions remains a challenging problem. In similar problems, such as image captioning, the gold standard is human annotation, though for styled image captions there is no consensus on how to perform such evaluations. Even describing the target style to annotators is difficult, a problem compounded by the limited linguistic knowledge of many annotators on crowd-sourcing platforms. Standardising approaches to human evaluation of styled captions, as done in descriptive image captioning [Chen et al., 2015], is an important direction for future work. Automatic evaluations for styled captions are also necessary as they can be run during development and have lower experimental variability than human evaluators. I suggested automatic metrics in Chapter 6, but more work needs to be done to standardise the approach and to develop metrics that simultaneously measure style and content.

Learning to extract semantic terms. The SemStyle system in Chapter 6 uses a discrete intermediate semantic representation defined by a set of rules. This was necessary to ensure the intermediate representation would apply to any style dataset, and to allow content planning decisions to be made by the visual components of the model. This intermediate representation may not adapt well to other semantic domains, or improved classifier accuracy. For example, the decision to exclude adjectives from the intermediate representation was influenced by the poor attribute identification performance of CNN+RNN caption generators [Vinyals et al., 2017]. Stronger attribute identification performance would necessitate inclusion in the intermediate representation to preserve semantic relevance. Ideally, each term in this semantic representation is mapped to its equivalence class, defined by both genera semantic

similarity and by the capacity of the vision system. For example if “*fury*” cannot be visually separated from “*hairy*”, they should be part of the same class even though in general they have different semantics.

Learning style attributes from document collections. In this thesis I have considered two approaches for defining style, with an attribute such as sentiment polarity, or with a document collection such as romance novels. An alternative is to learn a collection of style attributes from a collection of documents in different styles – an approach which bears resemblance to topic models. Doing so would open up some interesting avenues in terms of clustering styles and interpreting them for users. Such discrete style attributes could also be used to summarise the style of a document collection, potentially requiring fewer documents to fit.

Automatic source code commenting. A related task to image captioning is automatic source code commenting [Wong et al., 2015; Iyer et al., 2016], where comments for source code are generated automatically. Software development teams often define best practices for commenting source code. Differences in best practice can be considered stylistic. Ideally, an automatic source code captioning system would fit the style used in the rest of the code, as a consistent style is known to ease communication within teams. The main differences between this task and image captioning is source of the semantics being structured text rather than images, and the highly contextual nature of source code.

7.3 Final Remarks

Stylized text generation and image understanding both have many remaining challenges. For instance, skilled human authors can use targeted style choices to encourage a response from the reader, while automated systems are not yet at this level. If automatic stylized caption generation is to reach this level we need accurate models for the effect of style on the reader, and consistent methods for introducing style while preserving meaning. This thesis makes some progress towards the latter, but both remain open problems. In regard to image understanding, human viewers are able to accurately identify many different visual concepts and parse complex scenes at different levels of granularity. Models for image understanding are improving, but lack the higher level reasoning required to fully understand complex scenes. With interest from both academia and industry, more advancements in style generation and image understanding are likely; however, it is not yet clear if these problems can be completely solved without the development of general artificial agents. Nonetheless, automated systems that understand the world and communicate their conclusions to us through clear and attractive language remain an enticing prospect worthy of further study.

Bibliography

- ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M.; AND OTHERS, 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Symposium on Operating Systems Design and Implementation*, vol. 16, 265–283. (cited on page 14)
- ABDEL-HAMID, O.; MOHAMED, A.-R.; JIANG, H.; DENG, L.; PENN, G.; AND YU, D., 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22, 10 (2014), 1533–1545. <https://ieeexplore.ieee.org/document/6857341>. (cited on page 16)
- ABDEL-HAMID, O.; MOHAMED, A.-R.; JIANG, H.; AND PENN, G., 2012. Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference On*, 4277–4280. IEEE. <https://ieeexplore.ieee.org/abstract/document/6288864/>. (cited on page 16)
- ABE, M.; NAKAMURA, S.; SHIKANO, K.; AND KUWABARA, H., 1988. Voice conversion through vector quantization. In *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, 655–658 vol.1. doi:10.1109/ICASSP.1988.196671. (cited on page 42)
- AMES, M. AND NAAMAN, M., 2007. Why We Tag: Motivations for Annotation in Mobile and Online Media. *Special Interest Group on Computer-Human Interaction*, (2007). (cited on page 99)
- ANDERSON, P.; FERNANDO, B.; JOHNSON, M.; AND GOULD, S., 2016. SPICE: Semantic Propositional Image Caption Evaluation. *European Conference on Computer Vision*, 1 (2016), 382–398. doi:10.1007/978-3-319-46454-1. <http://www.panderson.me/images/SPICE.pdf>. (cited on pages 34, 36, 146, 152, 153, 161, and 168)
- ANDERSON, P.; FERNANDO, B.; JOHNSON, M.; AND GOULD, S., 2017. Guided Open Vocabulary Image Captioning with Constrained Beam Search. *Conference on Empirical Methods in Natural Language Processing*, (2017). <http://aclweb.org/anthology/D17-1098>. (cited on pages 84 and 180)

-
- ANDREW SHIN, Y. U. AND HARADA, T., 2016. Image Captioning with Sentiment Terms via Weakly-Supervised Sentiment Dataset. In *Proceedings of the British Machine Vision Conference (BMVC)*, 53.1–53.12. BMVA Press. doi:10.5244/C.30.53. <https://dx.doi.org/10.5244/C.30.53>. (cited on pages 53 and 55)
- ANDREWS, S.; TSOCHANTARIDIS, I.; AND HOFMANN, T., 2003. Support Vector Machines for Multiple-Instance Learning. In *Advances in Neural Information Processing Systems 15*, 561–568. (cited on page 29)
- ANDRYCHOWICZ, M.; DENIL, M.; GOMEZ, S.; HOFFMAN, M. W.; PFAU, D.; SCHAUL, T.; SHILLINGFORD, B.; AND DE FREITAS, N., 2017. Learning to Learn without Gradient Descent by Gradient Descent. *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, (2017). doi:10.1007/s10115-008-0151-5. <http://arxiv.org/abs/1606.04474>. (cited on page 22)
- ANNE HENDRICKS, L.; VENUGOPALAN, S.; ROHRBACH, M.; MOONEY, R.; SAENKO, K.; DARRELL, T.; MAO, J.; HUANG, J.; TOSHEV, A.; CAMBURU, O.; AND OTHERS, 2016. Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. http://openaccess.thecvf.com/content_cvpr_2016/html/Hendricks_Deep_Compositional_Captioning_CVPR_2016_paper.html. (cited on pages 84 and 180)
- ARCHER, K. J. AND KIMES, R. V., 2008. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, (2008). (cited on page 63)
- ARGAMON, S. AND LEVITAN, S., 2005. Measuring the usefulness of function words for authorship attribution. In *Proceedings of the Joint Annual Conference of The Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, 1–3. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.6935>. (cited on pages 45 and 148)
- ARGAMON, S.; ŠARIĆ, M.; AND STEIN, S., 2003. Style mining of electronic messages for multiple authorship discrimination: first results. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2003), 475–480. doi:10.1145/956750.956805. <http://dl.acm.org/citation.cfm?id=956805>. (cited on pages 45 and 148)
- ARJOVSKY, M.; CHINTALA, S.; AND BOTTOU, L., 2017. Wasserstein GAN. *International Conference on Machine Learning*, (2017). (cited on page 47)

-
- AULI, M. AND RUSH, A. M., 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (2016). (cited on pages 25 and 128)
- BACCHIANI, M. AND ROARK, B., 2003. Unsupervised language model adaptation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference On*, vol. 1, I—I. IEEE. (cited on page 98)
- BACCIANELLA, S.; ESULI, A.; AND SEBASTIANI, F., 2010. SentiWordNet 3.0 : An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining SentiWordNet. *Analysis*, 0 (2010), 1–12. doi:10.1.1.61.7217. <http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf>. (cited on page 99)
- BAHDANAU, D.; CHO, K.; AND BENGIO, Y., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representations*, (2014). (cited on pages 25 and 26)
- BAKER, C. F.; FILLMORE, C. J.; AND LOWE, J. B., 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics* -, vol. 1, 86. Association for Computational Linguistics. doi:10.3115/980845.980860. <http://portal.acm.org/citation.cfm?doid=980451.980860>. (cited on page 149)
- BALAHUR, A. AND STEINBERGER, R., 2009. Rethinking Sentiment Analysis in the News: from Theory to Practice and back. *Proceedings of the '1st Workshop on Opinion Mining and Sentiment Analysis*, (2009), 1–12. <http://emm.jrc.it/overview.html><http://langtech.jrc.it/Documents/09{ }WOMSA-WS-Sevilla{ }Sentiment-Def{ }printed.pdf>. (cited on page 39)
- BARSALOU, L. W., 1982. Context-independent and context-dependent information in concepts. *Memory & Cognition*, 10, 1 (1982), 82–93. doi:10.3758/BF03197629. <http://www.springerlink.com/index/10.3758/BF03197629>. (cited on pages 58, 59, and 76)
- BASTIEN, F.; LAMBLIN, P.; PASCANU, R.; BERGSTRÄ, J.; GOODFELLOW, I. J.; BERGERON, A.; BOUCHARD, N.; AND BENGIO, Y., 2012. Theano: new features and speed improvements. *Advances in Neural Information Processing Systems*, (2012). (cited on pages 14 and 111)
- BAWARSHI, A. S. AND MARY JO REIFFE, 2010. *Genre: An Introduction to History, Theory, Research, and Pedagogy*. ISBN 9781602351714. doi:10.1016/j.ymthe.2004.03.015. (cited on page 38)

-
- BAYER, J.; WIERSTRA, D.; TOGELIUS, J.; AND SCHMIDHUBER, J., 2009. Evolving Memory Cell Structures for Sequence Learning. In *Artificial Neural Networks – ICANN 2009*, 755–764. Springer Berlin Heidelberg, Berlin, Heidelberg. (cited on page 24)
- BEBOUT, L., 1993. Clinical Linguistics & Phonetics Processing of negative morphemes in aphasia: An example of the complexities of the closed class/ open class concept. *Clinical Linguistics & Phonetics*, 7, 2 (1993), 161–172. doi:10.3109/02699209308985552doi.org/10.3109/02699209308985552. <http://www.tandfonline.com/action/journalInformation?journalCode=iclp20>. (cited on pages 3 and 43)
- BENGIO, S.; VINYALS, O.; JAITLEY, N.; AND SHAZEER, N., 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, (2015). doi:10.1201/9781420049176. http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2015_5956.pdf. (cited on page 156)
- BENGIO, Y.; DUCHARME, R.; VINCENT, P.; AND JAUVIN, C., 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, Feb (2003), 1137–1155. (cited on pages 5 and 11)
- BENGIO, Y. AND FRASCONI, P., 1994. Credit Assignment through Time: Alternatives to Backpropagation. In *Advances in Neural Information Processing Systems*, 75–82. (cited on page 15)
- BERG, A. C.; BERG, T. L.; DAUME, H.; DODGE, J.; GOYAL, A.; HAN, X.; MENSCH, A.; MITCHELL, M.; SOOD, A.; STRATOS, K.; AND YAMAGUCHI, K., 2012. Understanding and predicting importance in images. In *Computer Vision and Pattern Recognition*. (cited on pages 60, 61, 69, and 70)
- BERG, T. L.; BERG, A. C.; AND SHIH, J., 2010. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *Computer Vision – ECCV 2010*, 663–676. Springer Berlin Heidelberg, Berlin, Heidelberg. (cited on page 152)
- BERNARDI, R.; CAKICI, R.; ELLIOTT, D.; ERDEM, A.; ERDEM, E.; KELLER, F.; MUSCAT, A.; PLANK, B.; IKIZLER-CINBIS, N.; KELLER, F.; MUSCAT, A.; AND PLANK, B., 2016. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. *Journal of Artificial Intelligence Research*, 55 (2016), 409–442. doi:10.1613/jair.4900. <https://www.jair.org/media/4900/live-4900-9139-jair.pdf>. (cited on pages 18 and 28)
- BLOUS, F. R. AND KRAUSS, R. M., 1988. Dominance and accommodation in the conversational behaviours of same- and mixed-gender dyads. *Language & Com-*

-
- munication*, 8, 3-4 (jan 1988), 183–194. doi:10.1016/0271-5309(88)90016-X. <https://www.sciencedirect.com/science/article/pii/027153098890016X>. (cited on page 40)
- BIRD, H.; FRANKLIN, S.; AND HOWARD, D., 2002. ‘Little words’ - Not really: Function and content words in normal and aphasic speech. *Journal of Neurolinguistics*, 15, 3-5 (may 2002), 209–237. doi:10.1016/S0911-6044(01)00031-8. <http://www.sciencedirect.com/science/article/pii/S0911604401000318>. (cited on page 43)
- BIRD, S.; KLEIN, E.; AND LOPER, E., 2009. *Natural Language Processing with Python*, vol. 43. ISBN 9780596516499. doi:10.1097/00004770-200204000-00018. <http://www.amazon.com/dp/0596516495>. (cited on pages 64 and 157)
- BLEI, D. M.; EDU, B.; NG, A. Y.; EDU, A.; JORDAN, M. I.; AND EDU, J., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, (2003). doi:10.1162/jmlr.2003.3.4-5.993. (cited on page 43)
- BORTH, D.; JI, R.; CHEN, T.; BREUEL, T.; AND CHANG, S.-F., 2013. Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs. *ACM International Conference on Multimedia (MM)*, (2013), 223–232. doi:10.1145/2502081.2502282. <http://dl.acm.org/citation.cfm?doid=2502081.2502282>. (cited on pages 99 and 107)
- BOSSARD, L.; GUILLAUMIN, M.; AND VAN GOOL, L., 2014. Food-101 – Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision*, 446–461. http://www.vision.ee.ethz.ch/~lbossard/bossard_eccv14_food-101.pdf. (cited on page 12)
- BOTTOU, L., 1998. On-line learning and stochastic approximations. *On-Line Learning in Neural Networks*, 17, 9 (1998), 142. (cited on page 15)
- BOUREAU, Y.-L.; PONCE, J.; AND LECUN, Y., 2010. A Theoretical Analysis of Feature Pooling in Visual Recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 111–118. (cited on page 17)
- BOWMAN, S. R.; VILNIS, L.; VINYALS, O.; DAI, A. M.; JOZEFOWICZ, R.; AND BENGIO, S., 2016. Generating Sentences from a Continuous Space. *International Conference on Learning Representations*, (2016), 1–13. <http://arxiv.org/abs/1511.06349>. (cited on page 47)
- BROOKE, J. AND HIRST, G., 2013. A Multi-Dimensional Bayesian Approach to Lexical Style. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (2013), 673–679. <https://aclanthology.coli.uni-saarland.de/papers/N13-1078/n13-1078>. (cited on page 44)

- BROWN, P. E.; DELLA PIETRA, V. J.; DELLA PIETRA, S. A.; AND MERCER, R. L., 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Technical report. <http://www.aclweb.org/anthology/J93-2003>. (cited on page 125)
- BURTON, K.; JAVA, A.; SOBOROFF, I.; AND OTHERS, 2009. The ICWSM 2009 Spinn3r Dataset. In *Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*. (cited on page 44)
- CANNY, J., 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 6 (nov 1986), 679–698. doi:10.1109/TPAMI.1986.4767851. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767851>. (cited on page 13)
- CARLSON, K.; RIDDELL, A.; AND ROCKMORE, D., 2017. Zero-Shot Style Transfer in Text Using Recurrent Neural Networks. In *arXiv*, 1–18. <http://arxiv.org/abs/1711.04731>. (cited on page 49)
- CER, D.; YANG, Y.; KONG, S.-Y.; HUA, N.; LIMTIACO, N.; JOHN, R. S.; CONSTANT, N.; GUAJARDO-CESPEDES, M.; YUAN, S.; TAR, C.; AND OTHERS, 2018. Universal Sentence Encoder. *arXiv Preprint arXiv:1803.11175*, (2018). <https://arxiv.org/abs/1803.11175>. (cited on page 3)
- CHAIGNEAU, S. E.; BARSALOU, L. W.; AND ZAMANI, M., 2009. Situational information contributes to object categorization and inference. *Acta Psychologica*, 130, 1 (2009), 81–94. (cited on pages 7, 58, and 59)
- CHEN, L.; ZHANG, H.; XIAO, J.; NIE, L.; SHAO, J.; LIU, W.; AND CHUA, T.-S., 2017. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. *Computer Vision and Pattern Recognition*, (2017). doi:10.1109/CVPR.2017.667. http://openaccess.thecvf.com/content_cvpr_2017/papers/Chen_SCA-CNN_Spatial_and_CVPR_2017_paper.pdf. (cited on page 33)
- CHEN, T.; BORTH, D.; DARRELL, T.; AND CHANG, S.-F., 2014. DeepSentiBank: Visual Sentiment Concept Classification with Deep Convolutional Neural Networks. *arXiv Preprint arXiv:1410.8586*, (2014). <http://arxiv.org/abs/1410.8586>. (cited on pages 52 and 99)
- CHEN, X.; FANG, H.; LIN, T.-Y.; VEDANTAM, R.; GUPTA, S.; DOLLAR, P.; AND ZITNICK, C. L., 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv:1504.00325*, (2015). (cited on pages xxi, 32, 34, 35, 62, 103, 106, 109, 111, 112, 113, 146, 147, 157, 161, 168, and 180)

-
- CHENG, J. AND LAPATA, M., 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 484–494. Association for Computational Linguistics, Berlin, Germany. (cited on pages 123, 129, and 130)
- CHENG, Z.; CAVERLEE, J.; AND LEE, K., 2010. You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, 759–768. ACM, New York, NY, USA. doi:10.1145/1871437.1871535. <http://doi.acm.org/10.1145/1871437.1871535>. (cited on page 41)
- CHO, K.; VAN MERRIENBOER, B.; BAHDANAU, D.; AND BENGIO, Y., 2014a. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, (2014), 103–111. <http://arxiv.org/abs/1409.1259>. (cited on pages 23, 55, and 131)
- CHO, K.; VAN MERRIENBOER, B.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; AND BENGIO, Y., 2014b. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2014), 1724–1734. doi:10.3115/v1/D14-1179. <http://arxiv.org/abs/1406.1078>. (cited on pages 23 and 131)
- CHOLLET, F. AND OTHERS, 2015. Keras. <https://github.com/keras-team/keras>. (cited on page 14)
- CHUNG, J.; AHN, S.; AND BENGIO, Y., 2017. Hierarchical Multiscale Recurrent Neural Networks. *International Conference on Learning Representations*, (2017). (cited on pages 24 and 25)
- CHUNG, J.; GULCEHRE, C.; CHO, K.; AND BENGIO, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv*, (2014). <https://arxiv.org/abs/1412.3555>. (cited on page 131)
- CIREGAN, D.; MEIER, U.; AND SCHMIDHUBER, J., 2012. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, 3642–3649. doi:10.1109/CVPR.2012.6248110. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6248110. (cited on page 12)
- CIRESAN, D.; GIUSTI, A.; GAMBARDILLA, L. M.; AND SCHMIDHUBER, J., 2012. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In *Advances in Neural Information Processing Systems*, 2843–2851. (cited on page 14)

-
- COHN, T. AND LAPATA, M., 2008. Sentence Compression Beyond Word Deletion. *International Conference on Computational Linguistics*, (2008), 137–144. doi:10.3115/1599081.1599099. <http://www.aclweb.org/anthology/C08-1018>. (cited on page 124)
- COHN, T. AND LAPATA, M., 2009. Sentence Compression As Tree Transduction. *J. Artif. Int. Res.*, 34, 1 (apr 2009), 637–674. <http://dl.acm.org/citation.cfm?id=1622716.1622733>. (cited on pages 123 and 128)
- COLLINS, M., 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, July, 1–8. doi:10.3115/1118693.1118694. <http://www.aclweb.org/anthology/W02-1001>. (cited on page 149)
- COLLOBERT, R.; KAVUKCUOGLU, K.; AND FARABET, C., 2011. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*. (cited on page 14)
- COLTHEART, M., 1981. The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33, 4 (1981), 497–505. doi:10.1080/14640748108400805. (cited on pages 3 and 43)
- CONNEAU, A.; KIELA, D.; SCHWENK, H.; BARRAULT, L.; AND BORDES, A., 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *arXiv Preprint arXiv:1705.02364*, (2017). <https://arxiv.org/abs/1705.02364>. (cited on page 3)
- CORTES, C. AND VAPNIK, V., 1995. Support-Vector Networks. *Machine Learning*, 20, 3 (1995), 273–297. doi:10.1023/A:1022627411411. (cited on pages 13 and 70)
- COSTER, W. AND KAUCHAK, D., 2011. Learning to Simplify Sentences Using Wikipedia. *Workshop on Monolingual Text-To-Text Generation, ACL'11*, (2011). (cited on pages 125, 130, 135, 137, 138, and 139)
- DAI, W.; YANG, Q.; XUE, G.-R.; AND YU, Y., 2007. Boosting for Transfer Learning. In *Proceedings of the 24th International Conference on Machine Learning - ICML '07*, 193–200. doi:10.1145/1273496.1273521. <http://www.machinelearning.org/proceedings/icml2007/papers/72.pdf>. (cited on page 97)
- DALAL, N. AND TRIGGS, B., 2005. Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, 886–893. doi:10.1109/CVPR.2005.177. (cited on pages 12 and 13)

-
- DANESCU-NICULESCU-MIZIL, C.; GAMON, M.; AND DUMAIS, S., 2011. Mark my words! Linguistic Style Accommodation in Social Media. *Proceedings of the 20th International Conference on World Wide Web - WWW '11*, abs/1105.0, May (2011), 745. doi:10.1145/1963405.1963509. <http://portal.acm.org/citation.cfm?doid=1963405.1963509>. (cited on pages 2, 3, and 40)
- DAS, D.; CHEN, D.; MARTINS, A. F. T.; SCHNEIDER, N.; AND SMITH, N. A., 2014. Frame-Semantic Parsing. *Computational Linguistics*, 40, 1 (2014), 9–56. doi:10.1162/COLI_a_00163. http://www.mitpressjournals.org/doi/10.1162/COLI_a_00163. (cited on page 149)
- DAUMÉ III, H., 2007. Frustratingly Easy Domain Adaptation. *Annual Meeting of the Association for Computational Linguistics*, , June (2007), 256–263. <http://acl.ldc.upenn.edu/P/P07/P07-1033.pdf>. (cited on page 97)
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; AND FEI-FEI, L., 2009. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*. (cited on pages 58, 65, 67, and 72)
- DENKOWSKI, M. AND LAVIE, A., 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *EACL 2014 Workshop on Statistical Machine Translation*, 376–380. doi:10.1.1.675.6117. <http://www.aclweb.org/anthology/W/W14/W14-3348>. (cited on pages 34, 35, and 161)
- DER MAATEN, L. AND HINTON, G., 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, (2008). (cited on page 82)
- DEVLIN, J.; GUPTA, S.; GIRSHICK, R.; MITCHELL, M.; AND ZITNICK, C. L., 2015. Exploring Nearest Neighbor Approaches for Image Captioning. *arXiv Preprint arXiv:1505.04467*, (2015). <https://arxiv.org/abs/1505.04467>. (cited on page 28)
- DÍAZ-AGUDO, B.; GERVÁS, P.; AND GONZÁLEZ-CALERO, P. A., 2002. Poetry Generation in COLIBRI. In *Advances in Case-Based Reasoning*, 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg. (cited on page 46)
- DIELEMAN, S.; WILLETT, K. W.; AND DAMBRE, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450, 2 (2015), 1441–1459. (cited on page 14)
- DING, S. H. H.; FUNG, B. C. M.; IQBAL, F.; AND CHEUNG, W. K., 2016. Learning Stylometric Representations for Authorship Analysis. (2016). <http://arxiv.org/abs/1606.01219>. (cited on page 45)

- DIVVALA, S. K.; FARHADI, A.; AND GUESTRIN, C., 2014. Learning Everything about Anything: Webly-Supervised Visual Concept Learning. In *Computer Vision and Pattern Recognition*. (cited on pages 60 and 61)
- DONAHUE, J.; HENDRICKS, L. A.; GUADARRAMA, S.; ROHRBACH, M.; VENUGOPALAN, S.; SAENKO, K.; AND DARRELL, T., 2015. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *Computer Vision and Pattern Recognition*, (jun 2015). (cited on pages 5, 28, 32, 55, and 96)
- DONAHUE, J.; JIA, Y.; VINYALS, O.; HOFFMAN, J.; ZHANG, N.; TZENG, E.; AND DARRELL, T., 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *International Conference on Machine Learning*, 647–655. (cited on pages 14 and 20)
- DOYLE, G.; YUROVSKY, D.; AND FRANK, M. C., 2016. A Robust Framework for Estimating Linguistic Alignment in Twitter Conversations. In *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, 637–648. ACM Press, New York, New York, USA. doi:10.1145/2872427.2883091. <http://dl.acm.org/citation.cfm?doid=2872427.2883091>. (cited on pages 2, 3, and 40)
- EBESU HUBBARD, A. S., 2009. Perspective Taking, Adaptation, and Coordination. In *21st Century Communication: A Reference Handbook 21st Century Communication: A Reference Handbook*, 119–127. SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States. doi:10.4135/9781412964005.n14. <http://sk.sagepub.com/reference/communication/n14.xml>. (cited on page 40)
- EDMUNDSON, H. P., 1969. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16, 2 (1969), 264–285. (cited on page 129)
- EISENSTEIN, J.; O'CONNOR, B.; SMITH, N. A.; AND XING, E. P., 2010. A Latent Variable Model for Geographic Lexical Variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, 1277–1287. Association for Computational Linguistics, Stroudsburg, PA, USA. <http://dl.acm.org/citation.cfm?id=1870658.1870782>. (cited on page 41)
- EL HIHI, S. AND BENGIO, Y., 1996. Hierarchical Recurrent Neural Networks for Long-Term Dependencies. In *Advances in Neural Information Processing Systems*, 493–499. (cited on page 24)
- ELLIOTT, D.; FRANK, S.; BARRAULT, L.; BOUGARES, F.; AND SPECIA, L., 2017. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual

-
- Image Description. *Proceedings of the Conference on Machine Translation*, 2 (2017), 215–233. <http://www.statmt.org/wmt17/pdf/WMT18.pdf>. (cited on page 54)
- ELMAN, J. L., 1990. Finding Structure in Time. *Cognitive Science*, 14, 2 (1990), 179–211. doi:10.1016/0364-0213(90)90002-E. <https://crl.ucsd.edu/~elman/Papers/fsit.pdf>. (cited on pages 21 and 22)
- ERKAN, G. AND RADEV, D. R., 2004. LexRank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22 (2004), 457–479. (cited on page 129)
- ESULI, A. AND SEBASTIANI, F., 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. *Language Resources and Evaluation Conference*, (2006). (cited on pages 99 and 107)
- FANG, H.; GUPTA, S.; IANDOLA, F.; SRIVASTAVA, R. K.; DENG, L.; DOLLAR, P.; GAO, J.; HE, X.; MITCHELL, M.; PLATT, J. C.; LAWRENCE ZITNICK, C.; ZWEIG, G.; DOLLÁR, P.; GAO, J.; HE, X.; MITCHELL, M.; PLATT, J. C.; ZITNICK, C. L.; AND ZWEIG, G., 2015. From Captions to Visual Concepts and Back. *Computer Vision and Pattern Recognition*, (2015). <https://arxiv.org/pdf/1411.4952.pdf>. (cited on pages 18, 30, 31, 33, and 84)
- FARHADI, A.; HEJRATI, M.; SADEGHI, M. A.; YOUNG, P.; RASHTCHIAN, C.; HOCKENMAIER, J.; AND FORSYTH, D., 2010. Every picture tells a story: Generating sentences from images. *European Conference on Computer Vision*, (2010). (cited on pages 28 and 31)
- FELZENSZWALB, P. F.; GIRSHICK, R. B.; MCALLESTER, D.; AND RAMANAN, D., 2010. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 9 (sep 2010), 1627–1645. doi: 10.1109/TPAMI.2009.167. <http://ieeexplore.ieee.org/document/5255236/>. (cited on pages 13, 30, and 58)
- FENG, Y. AND LAPATA, M., 2010. How Many Words Is a Picture Worth? Automatic Caption Generation for News Images. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, , July (2010), 1239–1249. <http://portal.acm.org/citation.cfm?id=1858807>. (cited on page 2)
- FERNÁNDEZ, S.; GRAVES, A.; AND SCHMIDHUBER, J., 2007. Sequence labelling in structured domains with hierarchical recurrent neural networks. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007*. (cited on page 24)

-
- FICLER, J. AND GOLDBERG, Y., 2017. Controlling Linguistic Style Aspects in Neural Language Generation. *EMNLP Workshop on Stylistic Variation*, (2017), 94–104. <https://aclweb.org/anthology/W/W17/W17-4912.pdf>. (cited on pages 6, 22, 38, 46, and 180)
- FILIPPOVA, K.; ALFONSECA, E.; COLMENARES, C. A.; KAISER, L.; AND VINYALS, O., 2015. Sentence Compression by Deletion with LSTMs. *Empirical Methods in Natural Language Processing*, (2015), 360–368. (cited on page 128)
- FILIPPOVA, K. AND ALTUN, Y., 2013. Overcoming the Lack of Parallel Data in Sentence Compression. *Empirical Methods in Natural Language Processing*, (2013), 1481–1491. (cited on page 128)
- FISZMAN, M.; RINDFLESCH, T. C.; AND KILICOGU, H., 2004. Abstraction Summarization for Managing the Biomedical Research Literature. In *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics*, CLS '04, 76–83. Association for Computational Linguistics, Stroudsburg, PA, USA. <http://dl.acm.org/citation.cfm?id=1596431.1596442>. (cited on page 129)
- FU, Z.; TAN, X.; PENG, N.; ZHAO, D.; AND YAN, R., 2018. Style Transfer in Text: Exploration and Evaluation. *AAAI Conference on Artificial Intelligence*, (2018), 663–670. <http://arxiv.org/abs/1711.06861>. (cited on pages 2, 38, and 49)
- GAN, C.; GAN, Z.; HE, X.; GAO, J.; AND DENG, L., 2017a. StyleNet: Generating Attractive Visual Captions with Styles. *Computer Vision and Pattern Recognition*, (2017). https://zhengan27.github.io/Papers/StyleNet_CVPR2017.pdf. (cited on pages xix, 3, 4, 51, 55, 157, 158, 162, and 166)
- GAN, Z.; GAN, C.; HE, X.; PU, Y.; TRAN, K.; GAO, J.; CARIN, L.; DENG, L.; AND UNIVERSITY, D., 2017b. Semantic Compositional Networks for Visual Captioning. *Computer Vision and Pattern Recognition*, (2017). <https://arxiv.org/pdf/1611.08002.pdf>. (cited on pages 33, 84, and 157)
- GANITKEVITCH, J.; VAN DURME, B.; AND CALLISON-BURCH, C., 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 758–764. <http://www.aclweb.org/anthology/N13-1092>. (cited on page 127)
- GAO, J.; FAN, W.; JIANG, J.; AND HAN, J., 2008. Knowledge transfer via multiple model local structure mapping. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08*, 283. ACM Press, New

-
- York, New York, USA. doi:10.1145/1401890.1401928. <http://dl.acm.org/citation.cfm?doid=1401890.1401928>. (cited on page 98)
- GATT, A. AND REITER, E., 2009. SimpleNLG: A Realisation Engine for Practical Applications. *European Workshop on Natural Language Generation*, (2009), 90–93. <http://aclweb.org/anthology/W/W09/W09-0613.pdf>. (cited on pages 21 and 30)
- GATYS, L. A.; ECKER, A. S.; AND BETHGE, M., 2016. Image Style Transfer Using Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference On*, 2414–2423. IEEE. (cited on pages 41 and 42)
- GERS, F. AND SCHMIDHUBER, J., 2000. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 189–194 vol.3. doi:10.1109/IJCNN.2000.861302. <http://ieeexplore.ieee.org/document/861302/>. (cited on page 23)
- GERS, F. A.; SCHMIDHUBER, J.; AND CUMMINS, F., 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12, 10 (2000), 2451–2471. doi:10.1162/089976600300015015. <http://www.mitpressjournals.org/doi/10.1162/089976600300015015>. (cited on page 23)
- GERVÁS, P., 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*, 14, 3-4 (2001), 181–188. (cited on page 46)
- GHAZVININEJAD, M.; SHI, X.; CHOI, Y.; AND KNIGHT, K., 2016. Generating Topical Poetry. *Empirical Methods in Natural Language Processing*, (2016), 1183–1191. <https://www.isi.edu/natural-language/mt/generating-topical-poetry.pdf>. (cited on page 47)
- GIBIANSKY, A.; ARIK, S.; DIAMOS, G.; MILLER, J.; PENG, K.; PING, W.; RAIMAN, J.; AND ZHOU, Y., 2017. Deep Voice 2: Multi-Speaker Neural Text-to-Speech. *Advances in Neural Information Processing Systems 30*, (2017). <https://papers.nips.cc/paper/6889-deep-voice-2-multi-speaker-neural-text-to-speech>. (cited on pages 41 and 42)
- GILDEA, D. AND SATTÀ, G., 2016. Synchronous Context-Free Grammars and Optimal Parsing Strategies. *Computational Linguistics*, 42, 2 (2016). doi:10.1162/COLI. <http://www.aclweb.org/anthology/J16-2002>. (cited on page 126)
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; AND MALIK, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2014), 580–587. doi:10.1109/CVPR.2014.81. (cited on page 29)

- GLOROT, X. AND BENGIO, Y., 2010. Understanding the difficulty of training deep feed-forward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9 (2010), 249–256. doi:10.1.1.207.2059. <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf>. (cited on pages 14 and 19)
- GLOROT, X.; BORDES, A.; AND BENGIO, Y., 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. *Proceedings of the 28th International Conference on International Conference on Machine Learning*, (2011), 513–520. <https://dl.acm.org/citation.cfm?id=3104547>. (cited on page 98)
- GOODFELLOW, I.; BENGIO, Y.; AND COURVILLE, A., 2016. *Deep Learning*. MIT Press. (cited on pages 15, 17, and 19)
- GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; AND BENGIO, Y., 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2672–2680. (cited on pages 47 and 48)
- GRAVES, A., 2013. Generating Sequences With Recurrent Neural Networks. (2013), 1–43. doi:10.1145/2661829.2661935. <http://arxiv.org/abs/1308.0850>. (cited on pages 5, 11, 23, 24, 55, and 112)
- GRAVES, A.; LIWICKI, M.; FERNÁNDEZ, S.; BERTOLAMI, R.; BUNKE, H.; AND SCHMIDHUBER, J., 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 5 (2009), 855–868. (cited on page 24)
- GRAVES, A. AND SCHMIDHUBER, J., 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, 2047–2052. doi:10.1109/IJCNN.2005.1556215. (cited on page 23)
- GREFF, K.; SRIVASTAVA, R. K.; KOUTNIK, J.; STEUNEBRINK, B. R.; AND SCHMIDHUBER, J., 2016. LSTM: A Search Space Odyssey. doi:10.1109/TNNLS.2016.2582924. <https://arxiv.org/pdf/1503.04069.pdf>. (cited on pages 23 and 24)
- GULRAJANI, I.; AHMED, F.; ARJOVSKY, M.; DUMOULIN, V.; AND COURVILLE, A. C., 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 5769–5779. (cited on pages 47 and 48)
- GUPTA, A. AND MANNEM, P., 2012. From Image Annotation to Image Description. *Advances in Neural Information Processing Systems*, (2012). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.6593&rep=rep1&type=pdf>. (cited on pages 29 and 30)

-
- HA, D.; DAI, A.; AND LE, Q. V., 2017. HyperNetworks. *International Conference on Learning Representations*, (2017). <https://arxiv.org/abs/1609.09106>. (cited on page 21)
- HAN, M.; WU, O.; AND NIU, Z., 2018. Unsupervised Automatic Text Style Transfer using LSTM. <http://tcci.ccf.org.cn/conference/2017/papers/1135.pdf>. (cited on pages 48 and 50)
- HATZIVASSILOPOULOS, V. AND MCKEOWN, K. R., 1997. Predicting the Semantic Orientation of Adjectives. *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics -*, (1997), 174–181. doi:10.3115/979617.979640. <http://portal.acm.org/citation.cfm?doid=979617.979640>. (cited on page 99)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *International Conference on Computer Vision*, (2015). doi:10.1.1.725.4861. https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/He_Delving_Deep_into_ICCV_2015_paper.pdf. (cited on pages 13 and 19)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. doi:10.1109/CVPR.2016.90. <http://ieeexplore.ieee.org/document/7780459/>. (cited on pages 5, 11, 14, 18, 19, 54, and 158)
- HEAFIELD, K.; POZYREVSKY, I.; CLARK, J. H.; AND KOEHN, P., 2013. Scalable Modified Kneser-Ney Language Model Estimation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, (2013), 690–696. https://kheafield.com/papers/edinburgh/estimate_paper.pdf. (cited on page 161)
- HEINE, S. J. AND LEHMAN, D. R., 1995. Cultural variation in unrealistic optimism: Does the West feel more vulnerable than the East? *Journal of Personality and Social Psychology*, 68, 4 (1995), 595–607. doi:10.1037/0022-3514.68.4.595. (cited on pages 115 and 116)
- HERMANN, K. M.; KOČISKÝ, T.; GREFFENSTETTE, E.; ESPEHOLT, L.; KAY, W.; SULEYMAN, M.; AND BLUNSON, P., 2015. Teaching machines to read and comprehend. <https://dl.acm.org/citation.cfm?id=2969428>. (cited on page 129)
- HOCHREITER, S. AND SCHMIDHUBER, J., 1997. Long Short-Term Memory. *Neural Computation*, 9, 8 (1997), 1735–1780. (cited on page 23)

-
- HODGKIN, A. L. AND HUXLEY, A. F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117, 4 (1952), 500–544. (cited on page 14)
- HODOSH, M.; YOUNG, P.; AND HOCKENMAIER, J., 2013. Framing image description as a ranking task: data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47 (2013), 853–899. doi:10.1613/jair.3994. <https://www.jair.org/media/3994/live-3994-7274-jair.pdf>. (cited on pages 28, 29, 32, 60, 106, and 110)
- HORN, C.; MANDUCA, C.; AND KAUCHAK, D., 2014. Learning a Lexical Simplifier Using Wikipedia. *Annual Meeting of the Association for Computational Linguistics*, (2014), 458–463. <http://www.aclweb.org/anthology/P/P14/P14-2075>. (cited on pages 128 and 130)
- Hovy, E. H., 2015. *What are Sentiment, Affect, and Emotion? Applying the Methodology of Michael Zock to Sentiment Analysis*, 13–24. Springer International Publishing, Cham. ISBN 978-3-319-08043-7. doi:10.1007/978-3-319-08043-7_2. https://doi.org/10.1007/978-3-319-08043-7_{_}2. (cited on pages 38 and 39)
- HU, J.; HUJIE, M.; LI SHEN, M.; AND SUN, G., 2017a. Squeeze-and-Excitation Networks. *Computer Vision and Pattern Recognition*, (2017). <https://arxiv.org/pdf/1709.01507.pdf>. (cited on pages 18 and 20)
- HU, Z.; YANG, Z.; LIANG, X.; SALAKHUTDINOV, R.; AND XING, E. P., 2017b. Controllable Text Generation. *arXiv Preprint arXiv:1703.00955*, (2017). (cited on page 48)
- HU, Z.; YANG, Z.; LIANG, X.; SALAKHUTDINOV, R.; AND XING, E. P., 2017c. Toward Controlled Generation of Text. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, 1587–1596. PMLR, International Convention Centre, Sydney, Australia. <http://proceedings.mlr.press/v70/hu17e.html>. (cited on page 47)
- HUFFAKER, D. A.; SWAAB, R.; AND DIERMEIER, D., 2011. The Language of Coalition Formation in Online Multiparty Negotiations. *Journal of Language and Social Psychology*, 30, 1 (2011), 66–81. doi:10.1177/0261927X10387102. <http://jls.sagepub.com>. (cited on page 40)
- IOFFE, S. AND SZEGEDY, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, (2015). doi:10.1007/s13398-014-0173-7.2. <http://proceedings.mlr.press/v37/ioffe15.pdf>. (cited on pages 18 and 19)

-
- IYER, S.; KONSTAS, I.; CHEUNG, A.; AND ZETTLEMOYER, L., 2016. Summarizing Source Code using a Neural Attention Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2073–2083. <http://www.aclweb.org/anthology/P16-1195>. (cited on page 181)
- JAFFE, A., 2017. Generating Image Descriptions using Multilingual Data. *Proceedings of the Conference on Machine Translation*, 2 (2017), 458–464. <https://apjaffe.github.io/jaffe2017multimodal.pdf>. (cited on page 54)
- JHAMTANI, H.; GANGAL, V.; HOVY, E.; AND NYBERG, E., 2017. Shakespearizing Modern Language Using Copy-Enriched Sequence-to-Sequence Models. *EMNLP Workshop on Stylistic Variation*, (2017), 10–19. <http://arxiv.org/abs/1707.01161>. (cited on pages 46 and 50)
- JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; AND DARRELL, T., 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv Preprint arXiv:1408.5093*, (2014). (cited on pages 14, 58, 60, 67, and 73)
- JIN, J. AND NAKAYAMA, H., 2016. Annotation Order Matters: Recurrent Image Annotator for Arbitrary Length Image Tagging. (2016). <https://arxiv.org/pdf/1604.05225.pdf>. (cited on page 32)
- JIN, O.; LIU, N. N.; ZHAO, K.; YU, Y.; AND YANG, Q., 2011. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11*, 775. ACM Press, New York, New York, USA. doi:10.1145/2063576.2063689. <http://dl.acm.org/citation.cfm?doid=2063576.2063689>. (cited on page 43)
- JING, H., 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing, ANLC '00*, 310–315. Association for Computational Linguistics, Stroudsburg, PA, USA. doi:10.3115/974147.974190. <https://doi.org/10.3115/974147.974190>. (cited on page 128)
- JOACHIMS, T., 2002. Optimizing search engines using clickthrough data. In *Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining*. (cited on page 71)
- JOACHIMS, T., 2006. Training Linear SVMs in Linear Time. In *Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining*. (cited on page 73)

-
- JOHANSSON, R. AND NUGUES, P., 2007. LTH: Semantic Structure Extraction using Nonprojective Dependency Trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 227–230. Association for Computational Linguistics. <http://www.aclweb.org/anthology/S07-1048>. (cited on page 149)
- JOHNSON, M.; SCHUSTER, M.; LE, Q. V.; KRIKUN, M.; WU, Y.; CHEN, Z.; THORAT, N.; VIÉGAS, F.; WATTENBERG, M.; CORRADO, G.; HUGHES, M.; AND DEAN, J., 2017. Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5 (2017), 339–351. <https://transacl.org/ojs/index.php/tacl/article/view/1081>. (cited on pages 49 and 156)
- JOHNSTONE, B., 1999. Uses of Southern-sounding speech by contemporary Texas women. *Journal of Sociolinguistics*, 3, 4 (1999), 505–522. doi:10.1111/1467-9481.00093. <http://dx.doi.org/10.1111/1467-9481.00093>. (cited on page 41)
- JOLICOEUR, P.; GLUCK, M. A.; AND KOSSLYN, S. M., 1984. Pictures and names: Making the connection âŸŸ. *Cognitive Psychology*, 16, 2 (1984). <http://www.ncbi.nlm.nih.gov/pubmed/6734136>. (cited on pages 58, 59, and 61)
- JONES, K. S.; WALKER, S.; AND ROBERTSON, S. E., 2000. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management: An International Journal*, 36, 6 (nov 2000), 779–808. doi:10.1016/S0306-4573(00)00015-7. <http://www.sciencedirect.com/science/article/pii/S0306457300000157?via%3Dihub>. (cited on page 158)
- JOSHI, D.; DATTA, R.; FEDOROVSKAYA, E.; LUONG, Q.-T.; WANG, J. Z.; LI, J.; AND LUO, J., 2011. Aesthetics and Emotions in Images. *Signal Processing Magazine, IEEE*, (2011). (cited on page 99)
- JOU, B.; CHEN, T.; PAPPAS, N.; REDI, M.; TOPKARA, M.; AND CHANG, S.-F., 2015. Visual Affect Around the World: A Large-scale Multilingual Visual Sentiment Ontology. *Proceedings of the 23rd ACM International Conference on Multimedia*, , 1 (2015). doi:10.1145/2733373.2806246. <http://arxiv.org/abs/1508.03868>. (cited on pages 99 and 116)
- JOZEFOWICZ, R.; ZAREMBA, W.; AND SUTSKEVER, I., 2015. An Empirical Exploration of Recurrent Network Architectures. *Proceedings of the 32nd International Conference on Machine Learning*, 37 (2015), 2342–2350. doi:10.1109/CVPR.2015.7298761. <http://proceedings.mlr.press/v37/jozefowicz15.pdf>. (cited on pages 23 and 24)

-
- JURAFSKY, D. AND MARTIN, J. H., 2014. *Speech and Language Processing*, vol. 3. Pearson London:. (cited on page 21)
- KAIN, A. AND MACON, M. W., 1998. Spectral voice conversion for text-to-speech synthesis. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference On*, vol. 1, 285–288 vol.1. doi:10.1109/ICASSP.1998.674423. (cited on page 42)
- KALCHBRENNER, N. AND BLUNSON, P., 2013. Recurrent Convolutional Neural Networks for Discourse Compositionality. *Proceedings of the 2013 Workshop on Continuous Vector Space Models and Their Compositionality*, (2013). (cited on page 25)
- KARAYEV, S.; HERTZMANN, A.; WINNEMOELLER, H.; AGARWALA, A.; AND DARRELL, T., 2013. Recognizing Image Style. *arXiv Preprint arXiv:1311.3715*, (2013). (cited on page 68)
- KARAYIL, T.; BLANDFORT, P.; BORTH, D.; AND DENGEL, A., 2016. Generating Affective Captions Using Concept And Syntax Transition Networks. In *Proceedings of the 2016 ACM on Multimedia Conference, MM '16*, 1111–1115. ACM, New York, NY, USA. doi:10.1145/2964284.2984070. <http://doi.acm.org/10.1145/2964284.2984070>. (cited on pages 52 and 55)
- KARLGREN, J., 2005. The Wheres and Whyfores for Studying Textual Genre Computationally. *AAAI Fall Symposium on Style and Meaning in Language, Art and Music, Arlington, USA.*, (2005). <http://soda.swedish-ict.se/46/1/FSS804JKarlgren.pdf>. (cited on page 37)
- KARPATY, A. AND FEI-FEI, L., 2015. Deep Visual-Semantic Alignments for Generating Image Descriptions. *Computer Vision and Pattern Recognition*, (2015). (cited on pages 96 and 111)
- KARPATY, A.; JOULIN, A.; AND FEI-FEI, L., 2014a. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping. *Advances in Neural Information Processing Systems*, (2014), 1–9. <http://arxiv.org/abs/1406.5679>. (cited on pages 5, 28, 29, 32, and 55)
- KARPATY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; AND FEI-FEI, L., 2014b. Large-Scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732. http://openaccess.thecvf.com/content_cvpr_2014/papers/Karpathy_Large-scale_Video_Classification_2014_CVPR_paper.pdf. (cited on page 16)

- KESKAR, N. S.; MUDIGERE, D.; NOCEDAL, J.; SMELYANSKIY, M.; AND TANG, P. T. P., 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *International Conference on Learning Representations*, (2017). (cited on page 16)
- KHOSMOOD, F. AND LEVINSON, R. A., 2008. Automatic natural language style classification and transformation. <https://dl.acm.org/citation.cfm?id=2227980>. (cited on pages 37 and 38)
- KIM, Y.; JERNITE, Y.; SONTAG, D.; AND RUSH, A. M., 2016. Character-Aware Neural Language Models. In *AAAI Conference on Artificial Intelligence*, 2741–2749. (cited on page 21)
- KINGMA, D. P. AND BA, J. L., 2015. ADAM: A Method For Stochastic Optimization. *International Conference on Learning Representations*, (2015). (cited on pages 14, 15, 132, and 157)
- KINGMA, D. P. AND WELLING, M., 2013. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, , ML (2013), 1–14. doi:10.1051/0004-6361/201527329. <http://arxiv.org/abs/1312.6114>. (cited on page 47)
- KIROS, R., 2015. neural-storyteller. <https://github.com/ryankiros/neural-storyteller>. (cited on pages xix, 38, 51, 52, 55, and 158)
- KIROS, R.; SALAKHUTDINOV, R.; AND ZEMEL, R., 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *arXiv Preprint arXiv:1411.2539*, (2014), 1–13. <http://arxiv.org/abs/1411.2539>. (cited on pages 5, 28, 32, 52, 55, 158, 159, and 160)
- KIROS, R.; ZHU, Y.; SALAKHUTDINOV, R.; ZEMEL, R. S.; TORRALBA, A.; URTASUN, R.; AND FIDLER, S., 2015. Skip-Thought Vectors. *Advances in Neural Information Processing Systems*, , 786 (2015), 1–9. doi:10.1017/CBO9781107415324.004. <http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf>. (cited on pages 3, 52, and 159)
- KNESER, R. AND NEY, H., 1995. Improved backing-off for M-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference On*, vol. 1, 181–184. IEEE. (cited on page 21)
- KOEHN, P.; HOANG, H.; AND BIRCH, A., 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics*, , June (2007), 177–180. doi:10.3115/1557769.1557821. <http://dl.acm.org/citation.cfm?id=1557821>. (cited on pages 31, 46, 54, 125, 137, and 138)

-
- KOEHN, P.; OCH, F. J.; AND MARCU, D., 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*. doi:10.3115/1073445.1073462. <http://www.aclweb.org/anthology/N03-1017>. (cited on page 125)
- KOPPEL, M.; SCHLER, J.; AND BONCHEK-DOKOW, E., 2007. Measuring Differentiability: Unmasking Pseudonymous Authors. *Journal of Machine Learning Research*, 8 (2007), 1261–1276. <http://eprints.pascal-network.org/archive/00003399/>. (cited on page 45)
- KOPPEL, M. AND SHTRIMBERG, I., 2006. Good News or Bad News? Let the Market Decide. In *Computing Attitude and Affect in Text: Theory and Applications*, 297–301. Springer-Verlag, Berlin/Heidelberg. doi:10.1007/1-4020-4102-0_22. http://link.springer.com/10.1007/1-4020-4102-0_{_}22. (cited on page 39)
- KOUTNIK, J.; GREFF, K.; GOMEZ, F.; AND SCHMIDHUBER, J., 2014. A Clockwork RNN. In *International Conference on Machine Learning*, 1863–1871. (cited on page 24)
- KRISHNAMOORTHY, N.; MALKARNENKAR, G.; MOONEY, R.; SAENKO, K.; AND GUADARRAMA, S., 2013. Generating Natural-Language Video Descriptions Using Text-Mined Knowledge. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. <http://acl.eldoc.ub.rug.nl/mirror/W/W13/W13-13.pdf#page=20>. (cited on page 31)
- KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 1–9. doi:<http://dx.doi.org/10.1016/j.protcy.2014.09.007>. <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>. (cited on pages 5, 11, 13, 14, 18, 31, 54, 58, 60, and 70)
- KSHIRSAGAR, M.; THOMSON, S.; SCHNEIDER, N.; CARBONELL, J.; SMITH, N. A.; AND DYER, C., 2015. Frame-Semantic Role Labeling with Heterogeneous Annotations. *Proceedings of ACL*, (2015), 218–224. doi:10.3115/v1/P15-2036. <http://www.aclweb.org/anthology/P/P15/P15-2036.pdf>. (cited on page 149)
- KUCHAIEV, O. AND GINSBURG, B., 2017. Factorization tricks for LSTM networks. *ICLR 2017 Workshop*, (2017). <https://arxiv.org/abs/1703.10722>. (cited on page 21)
- KULKARNI, G.; PREMRAJ, V.; DHAR, S.; LI, S.; CHOI, Y.; BERG, A. C.; AND BERG, T. L., 2011. Baby talk: Understanding and generating simple image descriptions. *Computer Vision and Pattern Recognition*, (2011). (cited on pages 28, 30, and 31)

-
- KUZNETSOVA, P.; ORDONEZ, V.; BERG, A. C. A.; BERG, T. L.; AND CHOI, Y., 2012. Collective Generation of Natural Image Descriptions. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, (2012), 359–368. <http://dl.acm.org/citation.cfm?id=2390524.2390575><http://dl.acm.org/citation.cfm?id=2390575>. (cited on pages 29 and 30)
- KUZNETSOVA, P.; ORDONEZ, V.; BERG, T. L.; AND CHOI, Y., 2014. TREETALK: Composition and Compression of Trees for Image Descriptions. *Transactions of the Association for Computational Linguistics*, (2014). (cited on pages 29 and 30)
- LAKOFF, G., 1987. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, vol. 64. ISBN 0226468046. doi:10.2307/415440. <http://www.jstor.org/stable/415440?origin=crossref>. (cited on pages 7, 57, 78, 80, and 83)
- LE, Q. AND MIKOLOV, T., 2014. Distributed Representations of Sentences and Documents. In *International Conference on Machine Learning*, 1188–1196. <http://proceedings.mlr.press/v32/le14.html>. (cited on page 3)
- LEBRET, R.; PINHEIRO, P. O.; AND COLLOBERT, R., 2015. Simple Image Description Generator via a Linear Phrase-Based Approach. *International Conference on Learning Representations Workshop*, (dec 2015). <http://arxiv.org/abs/1412.8419>. (cited on pages 29 and 30)
- LECUN, Y.; BOTTOU, L.; BENGIO, Y.; AND HAFFNER, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2323. doi:10.1109/5.726791. <http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>. (cited on pages 12, 13, 16, and 18)
- LEE, S.-I.; CHATALBASHEV, V.; VICKREY, D.; AND KOLLER, D., 2007. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th International Conference on Machine Learning - ICML '07*, 489–496. ACM Press, New York, New York, USA. doi:10.1145/1273496.1273558. <http://portal.acm.org/citation.cfm?doid=1273496.1273558>. (cited on page 97)
- LERNER, J. S.; LI, Y.; VALDESOLO, P.; AND KASSAM, K. S., 2015. Emotion and Decision Making. *Psychology*, 66 (2015). (cited on pages 7 and 95)
- LI, J.; JIA, R.; HE, H.; AND LIANG, P., 2018. Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1 (2018), 1865–1874. doi:10.18653/v1/

-
- N18-1169. <http://aclweb.org/anthology/N18-1169><http://arxiv.org/abs/1804.06437>. (cited on pages 48 and 50)
- LI, J.; MONROE, W.; SHI, T.; RITTER, A.; AND JURAFSKY, D., 2017. Adversarial Learning for Neural Dialogue Generation. *arXiv Preprint arXiv:1701.06547*, (2017). (cited on pages 47 and 48)
- LI, S.; KULKARNI, G.; BERG, T. L.; BERG, A. C.; AND CHOI, Y., 2011. Composing Simple Image Descriptions using Web-scale N-grams. In *CoNLL '11 Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. (cited on pages 28, 30, and 31)
- LI, S.; OUYANG, Y.; WANG, W.; AND SUN, B., 2007. Multi-document summarization using support vector regression. In *Proceedings of DUC*. Citeseer. (cited on page 129)
- LIAO, X.; XUE, Y.; AND CARIN, L., 2005. Logistic Regression with an Auxiliary Data Source. *International Conference on Machine Learning*, (2005), 505–512. doi:10.1145/1102351.1102415. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.8036&rep=rep1&type=pdf>. (cited on page 97)
- LIN, C. Y., 2004. ROUGE: A Package for Automatic Evaluation of Summaries. *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, (2004), 25–26. (cited on pages 34, 35, and 137)
- LIN, R.; LIU, S.; YANG, M.; LI, M.; ZHOU, M.; AND LI, S., 2015. Hierarchical Recurrent Neural Network for Document Modeling. In *Empirical Methods in Natural Language Processing*, 899–907. (cited on pages 24 and 25)
- LIN, T.; HORNE, B. G.; TINO, P.; AND GILES, C. L., 1996. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7, 6 (1996), 1329–1338. (cited on page 24)
- LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; AND ZITNICK, C. L., 2014. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*, (2014), 740–755. (cited on page 61)
- LIU, F.; XIANG, T.; HOSPEDALES, T. M.; YANG, W.; AND SUN, C., 2016. Semantic Regularisation for Recurrent Image Annotation. (2016). doi:10.1109/CVPR.2017.443. <http://arxiv.org/abs/1611.05490>. (cited on page 33)
- LLORET, E. AND PALOMAR, M., 2009. A Gradual Combination of Features for Building Automatic Summarisation Systems. In *International Conference on Text, Speech and Dialogue*, 16–23. Springer. (cited on page 129)

- LLORET, E. AND PALOMAR, M., 2012. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37, 1 (2012), 1–41. (cited on page 129)
- LOPEZ, A. AND ADAM, 2008. Statistical machine translation. *ACM Computing Surveys*, 40, 3 (aug 2008), 1–49. doi:10.1145/1380584.1380586. <http://portal.acm.org/citation.cfm?doid=1380584.1380586>. (cited on pages 125 and 126)
- LOUPPE, G.; WEHENKEL, L.; SUTERA, A.; AND GEURTS, P., 2013. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*, 431–439. (cited on page 63)
- LOWE, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 2 (2004), 91–110. doi:10.1023/B:VISI.0000029664.99615.94. (cited on page 13)
- LU, C. AND TANG, X., 2015. Surpassing Human-Level Face Verification Performance on LFW with GaussianFace. *AAAI Conference on Artificial Intelligence*, (2015). <http://arxiv.org/abs/1404.3840>. (cited on page 12)
- LUDWIG, S.; DE RUYTER, K.; FRIEDMAN, M.; BRÜGGEN, E. C.; WETZELS, M.; AND PFANN, G., 2013. More Than Words: The Influence of Affective Content and Linguistic Style Matches in Online Reviews on Conversion Rates. *Journal of Marketing*, 77, 1 (jan 2013), 87–103. doi:10.1509/jm.11.0560. <http://journals.ama.org/doi/abs/10.1509/jm.11.0560>. (cited on pages 2, 7, and 40)
- LUHN, H. P., 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2, 2 (1958), 159–165. (cited on page 129)
- LUONG, M.-T.; LE, Q. V.; SUTSKEVER, I.; VINYALS, O.; AND KAISER, L., 2016. Multi-task Sequence to Sequence Learning. *International Conference on Learning Representations*, (2016). <http://arxiv.org/abs/1511.06114>. (cited on page 158)
- LUONG, M.-T.; PHAM, H.; AND MANNING, C. D., 2015. Effective Approaches to Attention-based Neural Machine Translation. *Empirical Methods in Natural Language Processing*, (2015). <http://arxiv.org/abs/1508.04025>. (cited on pages 26, 27, 124, 131, 134, and 154)
- LUONG, M.-T.; SUTSKEVER, I.; LE, Q. V.; VINYALS, O.; AND ZAREMBA, W., 2014. Addressing the Rare Word Problem in Neural Machine Translation. *arXiv*, (2014). doi:10.3115/v1/P15-1002. <http://arxiv.org/abs/1410.8206>. (cited on pages 27 and 134)

-
- MANI, I.; KLEIN, G.; HOUSE, D.; HIRSCHMAN, L.; FIRMIN, T.; AND SUNDHEIM, B., 2002. SUMMAC: a text summarization evaluation. *Natural Language Engineering*, 8, 1 (2002), 43–68. doi:10.1017/S1351324901002741. (cited on page 129)
- MANURUNG, H., 2004. An evolutionary algorithm approach to poetry generation. (2004). (cited on page 46)
- MANURUNG, H.; RITCHIE, G.; THOMPSON, H.; AND MANURUNG, H. M., 2000. Towards A Computational Model Of Poetry Generation Towards A Computational Model of Poetry Generation. Technical report, The University of Edinburgh. (cited on page 47)
- MAO, J.; XU, W.; YANG, Y.; WANG, J.; HUANGZHI, H.; AND YUILLE, A., 2015. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *International Conference on Learning Representations*, (2015). (cited on pages 5, 28, 32, 55, and 96)
- MARCU, D. AND WONG, W., 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, vol. 10, 133–139. doi:10.3115/1118693.1118711. <https://pdfs.semanticscholar.org/6871/cdae138742eb2021c30749ec19c895f5bd16.pdf><http://portal.acm.org/citation.cfm?doid=1118693.1118711>. (cited on page 125)
- MARESCHAL, D. AND TAN, S. H., 2007. Flexible and Context-Dependent Categorization by Eighteen-Month-Olds. *Child Development*, 78, 1 (2007), 19–37. (cited on pages 58, 59, 61, and 76)
- MASON, R. AND CHARNIAK, E., 2014. Nonparametric Method for Data-driven Image Captioning. In *Annual Meeting of the Association for Computational Linguistics*, 592–598. <http://www.aclweb.org/anthology/P14-2097>. (cited on pages 28 and 29)
- MATHEWS, A.; XIE, L.; AND HE, X., 2015. Choosing Basic-Level Concept Names Using Visual and Language Context. *Winter Conference on the Applications of Computer Vision (WACV)*, (2015). doi:10.1109/WACV.2015.85. (cited on page 31)
- MCCARGAR, V., 2004. Statistical Approaches to Automatic Text Summarization. *Bulletin of the Association for Information Science and Technology*, 30, 4 (2004), 21–25. (cited on page 129)
- McFARLAND, D. A.; RAMAGE, D.; CHUANG, J.; HEER, J.; MANNING, C. D.; AND JURAFSKY, D., 2013. Differentiating language usage through topic models. *Poetics*, 41, 6 (dec 2013), 607–625. doi:10.1016/j.poetic.2013.06.004. <https://www.sciencedirect.com/science/article/pii/S0304422X13000442>. (cited on page 43)

-
- MELAMUD, O.; GOLDBERGER, J.; AND DAGAN, I., 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. *Conference on Natural Language Learning*, (2016), 51–61. (cited on page 128)
- MENDENHALL, T., 1887. The Characteristic Curves of Composition. *American Association for the Advancement of Science*, (1887). (cited on page 44)
- MERITY, S.; XIONG, C.; BRADBURY, J.; AND SOCHER, R., 2016. Pointer Sentinel Mixture Models. *Advances in Neural Information Processing Systems*, (2016). doi:10.1109/ICASSP.2016.7472159. <http://arxiv.org/abs/1609.07843>. (cited on page 46)
- MIHALCEA, R. AND TARAU, P., 2004. TextRank: Bringing Order into Texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. (cited on page 129)
- MIKOLOV, T.; CHEN, K.; CORRADO, G.; AND DEAN, J., 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. (cited on pages 52, 71, 105, and 128)
- MIKOLOV, T.; KARAFIÁT, M.; BURGET, L.; CERNOCKÝ, J.; AND KHUDANPUR, S., 2010. Recurrent neural network based language model. *Interspeech*, (2010). (cited on pages 5, 11, 21, and 55)
- MILLER, G. A., 1995. WORDNET: A Lexical Database for English. *Commun. ACM*, 38, 11 (nov 1995). (cited on pages 62, 64, and 67)
- MISHKIN, D. AND MATAS, J., 2016. All you need is a good init. *International Conference on Learning Representations*, (nov 2016). <http://arxiv.org/abs/1511.06422>. (cited on page 19)
- MITCHELL, M.; DODGE, J.; GOYAL, A.; YAMAGUCHI, K.; STRATOS, K.; MENSCH, A.; BERG, A.; HAN, X.; BERG, T.; AND HEALTH, O., 2012. Midge: Generating Image Descriptions From Computer Vision Detections. *European Chapter of the Association for Computational Linguistics*, (2012), 747–756. <http://www.tamaraberg.com/papers/EACL12.pdf>. (cited on pages 30 and 31)
- MOAWAD, I. F. AND AREF, M., 2012. Semantic graph reduction approach for abstractive Text Summarization. In *2012 Seventh International Conference on Computer Engineering Systems (ICCES)*, 132–138. doi:10.1109/ICCES.2012.6408498. (cited on page 129)
- MÜLLER, T., 2009. The Uses of Authenticity: Hemingway and the Literary Field, 1926–1936. *Journal of Modern Literature*, 33, 1 (dec 2009), 28–42. doi:10.2979/

-
- JML.2009.33.1.28. <https://www.jstor.org/stable/10.2979/JML.2009.33.1.28>. (cited on page 123)
- MURPHY, K. P., 2012. *Machine Learning: A Probabilistic Perspective*. ISBN 0070428077. doi:10.1007/978-3-642-21004-4_10. (cited on page 15)
- MURRAY, N.; MARCHESOTTI, L.; AND PERRONNIN, F., 2012. AVA: A large-scale database for aesthetic visual analysis. *Computer Vision and Pattern Recognition*, (2012). (cited on page 99)
- NAIR, V. AND HINTON, G. E., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, 3 (2010), 807–814. doi:10.1.1.165.6419. (cited on pages 14 and 15)
- NAKAGAWA, T.; INUI, K.; AND KUROHASHI, S., 2010. Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables. *Computational Linguistics*, (2010). (cited on page 98)
- NALLAPATI, R.; ZHAI, F.; AND ZHOU, B., 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI Conference on Artificial Intelligence*, 3075–3081. (cited on pages 123, 129, and 130)
- NAPOLES, C.; CALLISON-BURCH, C.; AND POST, M., 2016. Sentential Paraphrasing as Black-Box Machine Translation. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016 (2016), 62–66. <http://aclweb.org/anthology/N16-3013>. (cited on page 180)
- NARAYAN, S. AND GARDENT, C., 2014. Hybrid Simplification using Deep Semantics and Machine Translation. *Annual Meeting of the Association for Computational Linguistics*, (2014), 435–445. (cited on page 138)
- NG, J. Y.-H.; HAUSKNECHT, M.; VIJAYANARASIMHAN, S.; VINYALS, O.; MONGA, R.; AND TODERICI, G., 2015. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference On*, 4694–4702. IEEE. (cited on pages 16 and 22)
- NIEDERHOFFER, K. G. AND PENNEBAKER, J. W., 2002. Linguistic Style Matching in Social Interaction. *Journal of Language and Social Psychology*, 21, 4 (2002), 337–360. doi:10.1177/026192702237953. <http://journals.sagepub.com/doi/10.1177/026192702237953>. (cited on page 40)

- NIELSEN, M. A., 2015. *Neural Networks and Deep Learning*. Determination Press. (cited on page 16)
- OBERLANDER, J. AND GILL, A. J., 2006. Language With Character: A Stratified Corpus Comparison of Individual Differences in E-Mail Communication. *Discourse Processes*, 42, 3 (2006), 239–270. http://www.tandfonline.com/doi/pdf/10.1207/s15326950dp4203_1?needAccess=true. (cited on pages 2 and 40)
- OCH, F. J.; TILLMANN, C.; AND NEY, H., 1999. Improved Alignment Models for Statistical Machine Translation. *Empirical Methods in Natural Language Processing*, (1999), 20–28. <http://www.aclweb.org/anthology/W99-0604>. (cited on page 125)
- OH, K.-S. AND JUNG, K., 2004. GPU implementation of neural networks. *Pattern Recognition*, 37, 6 (2004), 1311–1314. doi:<https://doi.org/10.1016/j.patcog.2004.01.013>. <http://www.sciencedirect.com/science/article/pii/S0031320304000524>. (cited on page 14)
- ORABY, S.; HOMAYON, S.; AND WALKER, M., 2017. Harvesting Creative Templates for Generating Stylistically Varied Restaurant Reviews. *EMNLP Workshop on Stylistic Variation*, (2017), 28–36. <http://aclweb.org/anthology/W17-4904>. (cited on pages 46 and 180)
- ORDONEZ, V.; DENG, J.; CHOI, Y.; BERG, A. C.; AND BERG, T. L., 2013. From Large Scale Image Categorization to Entry-Level Categories. In *International Conference on Computer Vision*. (cited on pages xx, xxiii, 7, 9, 58, 60, 61, 65, 66, 72, 73, 74, 76, 77, 85, and 86)
- ORDONEZ, V.; KULKARNI, G.; AND BERG, T. L., 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems*. <http://papers.nips.cc/paper/4470-im2text-describing-images-using-1-million-captioned-photographs.pdf>. (cited on pages 28, 32, 60, 72, and 79)
- PAETZOLD, G. AND SPECIA, L., 2016. SemEval 2016 Task 11: Complex Word Identification. *SemEval-2016*, (2016), 560–569. <http://aclweb.org/anthology/S16-1085>. (cited on pages 123 and 127)
- PAETZOLD, G. H. AND SPECIA, L., 2015. LEXenstein: A Framework for Lexical Simplification. *Annual Meeting of the Association for Computational Linguistics*, (2015), 85–90. (cited on page 128)
- PAN, S. J. AND YANG, Q., 2009. A Survey on Transfer Learning. *Ieee Transactions on Knowledge and Data Engineering*, (2009). doi:10.1109/TKDE.

- 2009.191. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.4126&rep=rep1&type=pdf>. (cited on page 97)
- PANG, B., 2006. *Opinion Mining and Sentiment Analysis*, vol. 1. ISBN 1601981503. doi: 10.1561/1500000001. <http://www.nowpublishers.com/product.aspx?product=INR&doi=1500000001>. (cited on pages 38 and 98)
- PANG, B. AND LEE, L., 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational*. www.cs.cornell.edu/people/pabo/movie-. (cited on page 39)
- PANG, B. AND LEE, L., 2008. *Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval*, (2008). (cited on pages 39 and 98)
- PAPINENI, K.; ROUKOS, S.; WARD, T.; AND ZHU, W., 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *Annual Meeting of the Association for Computational Linguistics*, (2002), 311–318. doi:10.3115/1073083.1073135. <http://dl.acm.org/citation.cfm?id=1073135>. (cited on pages 34, 137, and 161)
- PARDO, J. S.; GIBBONS, R.; SUPPES, A.; AND KRAUSS, R. M., 2012. Phonetic convergence in college roommates. *Journal of Phonetics*, 40, 1 (jan 2012), 190–197. doi:10.1016/J.WOCN.2011.10.001. <http://www.sciencedirect.com/science/article/pii/S0095447011000945>. (cited on page 40)
- PAULUS, R.; XIONG, C.; AND SOCHER, R., 2018. A Deep Reinforced Model for Abstractive Summarization. *International Conference on Learning Representations*, (may 2018). <http://arxiv.org/abs/1705.04304>. (cited on page 27)
- PAVALANATHAN, U. AND EISENSTEIN, J., 2015. Audience-Modulated Variation in Online Social Media. *American Speech*, 90, 2 (2015), 187–213. doi:10.1215/00031283-3130324. <http://americanspeech.dukejournals.org/content/90/2/187.full.pdf+html>. (cited on pages 40 and 179)
- PAVLICK, E. AND CALLISON-BURCH, C., 2013. Simple PPDB : A Paraphrase Database for Simplification. (2013). (cited on page 180)
- PAVLICK, E. AND CALLISON-BURCH, C., 2016. Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 143–148. <https://aclanthology.coli.uni-saarland.de/papers/P16-2024/p16-2024>. (cited on pages 125 and 127)

-
- PAVLICK, E. AND NENKOVA, A., 2015. Inducing Lexical Style Properties for Paraphrase and Genre Differentiation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 218–224. (cited on pages 6 and 38)
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; AND DUCHESNAY, É., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, Oct (2011), 2825–2830. <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>. (cited on page 64)
- PENNEBAKER, J. W.; CHUNG, C. K.; IRELAND, M.; GONZALES, A.; AND BOOTH, R. J., 2007. The Development and Psychometric Properties of LIWC2007. (2007), 1–22. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.600.7227&rep=rep1&type=pdf>. (cited on page 41)
- PENNEBAKER, J. W. AND KING, L. A., 1999. Linguistic styles : Language use as an individual difference. *Journal of Personality and Social Psychology*, 77, 6 (1999), 1296–1312. doi:10.1037/0022-3514.77.6.1296. <http://doi.apa.org/getdoi.cfm?doi=10.1037/0022-3514.77.6.1296>. (cited on pages 2, 37, and 40)
- PENNINGTON, J.; SOCHER, R.; AND MANNING, C. D., 2014. GloVe: Global Vectors for Word Representation. *Empirical Methods in Natural Language Processing*, (2014), 1532–1543. doi:10.3115/v1/D14-1162. (cited on page 132)
- PFISTER, T.; SIMONYAN, K.; CHARLES, J.; AND ZISSERMAN, A., 2014. Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos. In *Asian Conference on Computer Vision*, 538–552. Springer. (cited on page 20)
- PHILLIPS, P. J., 1999. Support Vector Machines Applied to Face Recognition. 11 (1999). http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=150750. (cited on page 13)
- PINTO, N.; COX, D. D.; AND DICARLO, J. J., 2008. Why is Real-World Visual Object Recognition Hard? *PLoS Computational Biology*, 4, 1 (2008), e27. doi:10.1371/journal.pcbi.0040027. <http://dx.plos.org/10.1371/journal.pcbi.0040027>. (cited on page 13)
- PLATT, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10, 3 (1999), 61–74. doi:10.1.1.41.1639. (cited on page 70)
- POPA, V.; NURMINEN, J.; AND GABBOUJ, M., 2009. A Novel Technique for Voice Conversion Based on Style and Content Decomposition with Bilinear Models. In *Tenth*

-
- Annual Conference of the International Speech Communication Association.* (cited on pages 4, 41, and 42)
- PRABHUMOYE, S.; TSVETKOV, Y.; SALAKHUTDINOV, R.; AND BLACK, A. W., 2018. Style Transfer Through Back-Translation. *Annual Meeting of the Association for Computational Linguistics*, (2018), 866–876. <http://aclweb.org/anthology/P18-1080><http://arxiv.org/abs/1804.09000>. (cited on pages 2, 38, 48, and 50)
- PRESS, O.; BAR, A.; BOGIN, B.; BERANT, J.; AND WOLF, L., 2017. Language Generation with Recurrent Generative Adversarial Networks without Pre-training. *Workshop on Learning to Generate Natural Language at ICML*, (2017). <http://arxiv.org/abs/1706.01399>. (cited on page 48)
- PRITCHARD, J. K.; STEPHENS, M.; AND DONNELLY, P., 2000. Inference of population structure using multilocus genotype data. *Genetics*, (2000). doi:10.1111/j.1471-8286.2007.01758.x. (cited on page 43)
- QIXIN, W.; LUO, T.; DONG, W.; AND CHAO, X., 2016. Chinese song iambics generation with neural attention-based model. In *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2016-Janua, 2943–2949. <https://pdfs.semanticscholar.org/221e/447e326dcc391eda417c48d696998bb0fbac.pdf>. (cited on page 47)
- RADFORD, A.; JÓZEFOWICZ, R.; AND SUTSKEVER, I., 2017. Learning To Generate Reviews and Discovering Sentiment. *CoRR*, abs/1704.0 (2017). <http://arxiv.org/abs/1704.01444>. (cited on page 41)
- RAINA, R.; BATTLE, A.; LEE, H.; PACKER, B.; AND NG, A. Y., 2007. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, 759–766. ACM. (cited on page 97)
- RAINA, R.; MADHAVAN, A.; AND NG, A. Y., 2009. Large-scale Deep Unsupervised Learning Using Graphics Processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 873–880. ACM, New York, NY, USA. doi:10.1145/1553374.1553486. <http://doi.acm.org/10.1145/1553374.1553486>. (cited on page 14)
- RAMSHAW, L. A. AND MARCUS, M. P., 1999. *Text Chunking Using Transformation-Based Learning*, 157–176. Springer Netherlands, Dordrecht. ISBN 978-94-017-2390-9. doi:10.1007/978-94-017-2390-9_10. https://doi.org/10.1007/978-94-017-2390-9_{ }10. (cited on page 118)

-
- RANZATO, M.; CHOPRA, S.; AULI, M.; AND ZAREMBA, W., 2015. Sequence Level Training with Recurrent Neural Networks. (2015), 1–16. doi:10.1371/journal.pcbi.1005055. <http://arxiv.org/abs/1511.06732>. (cited on page 33)
- RAO, S. AND TETREAULT, J., 2018. Dear Sir or Madam, May I introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. (2018), 129–140. <https://github.com/raosudha89/http://arxiv.org/abs/1803.06535>. (cited on page 50)
- RASHTCHIAN, C.; YOUNG, P.; HODOSH, M.; AND HOCKENMAIER, J., 2010. Collecting Image Annotations Using Amazon’s Mechanical Turk. <https://dl.acm.org/citation.cfm?id=1866717>. (cited on pages 30 and 32)
- RAZAVIAN, A. S.; AZIZPOUR, H.; SULLIVAN, J.; AND CARLSSON, S., 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 512–519. IEEE. doi:10.1109/CVPRW.2014.131. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6910029>. (cited on pages 20, 68, and 98)
- REITER, E.; SRIPADA, S.; HUNTER, J.; YU, J.; AND DAVY, I., 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167, 1 (2005), 137–169. doi:<https://doi.org/10.1016/j.artint.2005.06.006>. <http://www.sciencedirect.com/science/article/pii/S0004370205000998>. (cited on page 21)
- REN, S.; HE, K.; GIRSHICK, R.; AND SUN, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, (2015), 1–10. doi:10.1016/j.nima.2015.05.028. (cited on page 18)
- RENNIE, S. J.; MARCHERET, E.; MROUEH, Y.; ROSS, J.; AND GOEL, V., 2016. Self-Critical Sequence Training for Image Captioning. (2016). doi:10.1109/CVPR.2017.131. <http://arxiv.org/abs/1612.00563>. (cited on page 33)
- RINDFLESCH, T. C.; KILICOGLU, H.; FISZMAN, M.; ROSEMBLAT, G.; AND SHIN, D., 2011. Semantic MEDLINE: An advanced information management application for biomedicine. *Information Services & Use*, 31, 1-2 (2011), 15–21. (cited on page 129)
- ROBERTS, L. G., 1963. *Machine perception of three-dimensional solids*. Ph.D. thesis. <https://dspace.mit.edu/handle/1721.1/11589#files-area>. (cited on page 13)
- ROBINSON, A. J. AND FALLSIDE, F., 1987. *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering. (cited on page 24)

-
- ROHRBACH, M.; QIU, W.; TITOV, I.; THATER, S.; PINKAL, M.; AND SCHIELE, B., 2013. Translating Video Content to Natural Language Descriptions. *International Conference on Computer Vision*, (2013). doi:10.1109/ICCV.2013.61. http://www.cv-foundation.org/openaccess/content_iccv_2013/papers/Rohrbach_Translating_Video_Content_2013_ICCV_paper.pdf. (cited on page 31)
- ROSCH, E., 1999. *Principles of categorization*. MIT Press. (cited on pages 7, 57, and 59)
- ROSCH, E.; MERVIS, C. B.; GRAY, W. D.; JOHNSON, D. M.; AND BOYES-BRAEM, P., 1976. Basic objects in natural categories. *Cognitive Psychology*, 8, 3 (1976), 382–439. doi: [http://dx.doi.org/10.1016/0010-0285\(76\)90013-X](http://dx.doi.org/10.1016/0010-0285(76)90013-X). <http://www.sciencedirect.com/science/article/pii/001002857690013X>. (cited on pages 57, 58, 61, and 80)
- ROSENBLATT, F., 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, 65, 6 (1958), 386–408. doi:10.1037/h0042519. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775>. (cited on page 14)
- ROTH, M. AND LAPATA, M., 2015. Context-aware Frame-Semantic Role Labeling. *Transactions of the Association for Computational Linguistics*, 3 (2015), 449–460. <http://aclweb.org/anthology/Q15-1032>. (cited on page 149)
- RUMELHART, D. E.; HINTON, G. E.; AND WILLIAMS, R. J., 1986. Learning representations by back-propagating errors. *Nature*, 323, 6088 (1986), 533–536. doi: 10.1038/323533a0. (cited on page 15)
- RUSH, A. M.; CHOPRA, S.; AND WESTON, J., 2015. A Neural Attention Model for Abstractive Sentence Summarization. *Empirical Methods in Natural Language Processing*, (2015), 379–389. <http://arxiv.org/abs/1509.00685>. (cited on pages 25, 123, and 128)
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; AND FEI-FEI, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 3 (dec 2015), 211–252. doi:10.1007/s11263-015-0816-y. <http://link.springer.com/10.1007/s11263-015-0816-y>. (cited on pages 14, 73, and 153)
- RUSSELL, S. AND NORVIG, P., 2010. Artificial Intelligence: A Modern Approach, 3rd edition. *Prentice Hall*, (2010), 1 – 1132. doi:10.1017/S0269888900007724. (cited on page 13)

- SALIMANS, T.; GOODFELLOW, I.; ZAREMBA, W.; CHEUNG, V.; RADFORD, A.; AND CHEN, X., 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems*, 2234–2242. (cited on page 47)
- SALVETTI, F.; REICHENBACH, C.; AND LEWIS, S., 2006. Opinion Polarity Identification of Movie Reviews. *Computing Attitude and Affect in Text: Theory and Applications*, (2006), 303–316. (cited on page 99)
- SANTOS, C. D. AND GATTI, M., 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 69–78. <http://www.aclweb.org/anthology/C14-1008>. (cited on page 16)
- SANTOS, C. N. D.; MELNYK, I.; AND PADHI, I., 2018. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. *Annual Meeting of the Association for Computational Linguistics*, (2018), 189–194. <http://aclweb.org/anthology/P18-2031><http://arxiv.org/abs/1805.07685>. (cited on page 49)
- SCHERER, D.; MÜLLER, A.; AND BEHNKE, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, 92–101. Springer. (cited on page 17)
- SCHMIDHUBER, J., 1992. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4, 2 (1992), 234–242. (cited on page 24)
- SCHUSTER, M. AND PALIWAL, K., 1997a. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45, 11 (1997), 2673–2681. doi:10.1109/78.650093. <http://ieeexplore.ieee.org/document/650093/>. (cited on page 26)
- SCHUSTER, M. AND PALIWAL, K. K., 1997b. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45, 11 (1997), 2673–2681. doi:10.1109/78.650093. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=650093. (cited on page 154)
- SCHWEIKERT, G.; RÄTSCH, G.; WIDMER, C.; AND SCHÖLKOPF, B., 2008. An Empirical Analysis of Domain Adaptation Algorithms for Genomic Sequence Analysis. *Advances in Neural Information Processing Systems*, (2008). (cited on pages 97, 98, 103, and 113)
- SEE, A.; LIU, P. J.; AND MANNING, C. D., 2017. Get To The Point: Summarization with Pointer-Generator Networks. *Annual Meeting of the Association for Computational Linguistics*, (2017). (cited on page 130)

-
- SEE, A. AND MANNING, C. D., 2017. Get To The Point : Summarization with Pointer-Generator Networks. *Association for Computational Linguistics*, (2017), 1–18. <http://arxiv.org/abs/1704.04368>. (cited on page 143)
- SEIDE, F. AND AGARWAL, A., 2016. CNTK: Microsoft’s Open-Source Deep-Learning Toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2135. ACM. (cited on page 14)
- SEMIENIUTA, S.; SEVERYN, A.; AND BARTH, E., 2017. A Hybrid Convolutional Variational Autoencoder for Text Generation. *Empirical Methods in Natural Language Processing*, (2017). (cited on page 47)
- SHANNON, C. E., 1951. Prediction and Entropy of Printed English. *Bell Labs Technical Journal*, 30, 1 (1951), 50–64. (cited on page 21)
- SHAO, Y.; GOUWS, S.; BRITZ, D.; GOLDIE, A.; STROPE, B.; AND KURZWEIL, R., 2017. Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2210–2219. <http://www.aclweb.org/anthology/D/D17/D17-1235.pdf>. (cited on page 27)
- SHAROT, T., 2012. *The Optimism Bias: Why we’re wired to look on the bright side*. Hachette UK. (cited on page 115)
- SHEN, T.; LEI, T.; BARZILAY, R.; JAAKKOLA, T.; AND CSAIL, M., 2017. Style Transfer from Non-Parallel Text by Cross-Alignment. In *Advances in Neural Information Processing Systems*. <https://github.com/shentianxiao/language-style-transfer>. (cited on pages 2, 38, 48, 49, and 50)
- SHOEMARK, P.; KIRBY, J.; GOLDWATER, S.; AND UK, S. E. A., 2017a. Topic and audience effects on distinctively Scottish vocabulary usage in Twitter data. *EMNLP Workshop on Stylistic Variation*, (2017), 59–68. <https://aclweb.org/anthology/W/W17/W17-4908.pdf>. (cited on page 41)
- SHOEMARK, P.; SUR, D.; SHRIMPTON, L.; MURRAY, I.; GOLDWATER, S.; AND UK, S. E. A., 2017b. Aye or naw, whit dae ye hink? Scottish independence and linguistic identity on social media. *Conference of the European Chapter of the Association for Computational Linguistics*, 1 (2017), 1239–1248. <http://www.aclweb.org/anthology/E17-1116>. (cited on pages 41 and 179)
- SIMONYAN, K. AND ZISSERMAN, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, (2015). (cited on pages 5, 11, 18, 19, 20, 54, 79, 82, 104, 112, and 158)

-
- SIMPSON, P., 2004. *Stylistics: A Resource Book for Students*. Routledge. ISBN 0415281059. (cited on pages 2, 3, 37, and 57)
- SMITH, D. A. AND EISNER, J., 2006. Quasi-synchronous grammars: alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*, 23–30. Association for Computational Linguistics. <https://dl.acm.org/citation.cfm?id=1654655>. (cited on page 127)
- SMITH, S. L. AND LE, Q. V., 2018. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *International Conference on Learning Representations*, (2018). (cited on page 16)
- SNODGRASS, J. G. AND VANDERWART, M., 1980. A standardized set of 260 pictures: Norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of Experimental Psychology: Human Learning & Memory*, 6, 2 (1980), 174–215. doi:10.1037/0278-7393.6.2.174. (cited on page 59)
- SNOEK, J.; LAROCHELLE, H.; AND ADAMS, R. P., 2012. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*, (2012). (cited on page 106)
- SOCHER, R.; KARPATY, A.; LE, Q.; MANNING, C.; AND NG, A., 2014. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *Transactions of the Association for Computational Linguistics*, 2, 0 (2014), 207–218. <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/325>. (cited on page 29)
- SOCHER, R.; PERELYGIN, A.; WU, J. Y.; CHUANG, J.; MANNING, C. D.; NG, A. Y.; AND POTTS, C., 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Empirical Methods in Natural Language Processing*, (2013). (cited on page 98)
- SPAIN, M. AND PERONA, P., 2011. Measuring and Predicting Object Importance. *International Journal of Computer Vision*, 91 (2011), 59–76. doi:10.1007/s11263-010-0376-0. (cited on pages 60, 61, and 69)
- SPARCK JONES, K., 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 1 (1972), 11–21. (cited on page 36)
- SPECIA, L.; FRANK, S.; SIMA'AN, K.; AND ELLIOTT, D., 2016. A Shared Task on Multimodal Machine Translation and Crosslingual Image Description. *Proceedings of the First Conference on Machine Translation*, 2 (2016), 543–553. <http://www.statmt.org/wmt16/pdf/W16-2346.pdf>. (cited on page 54)

-
- SPECIA, L. AND JAUHAR, S. K., 2012. SemEval-2012 Task 1: English Lexical Simplification. *SemEval-2012*, (2012), 347–355. <http://newdesign.aclweb.org/anthology-new/S/S12/S12-1046.pdf>. (cited on pages 123 and 127)
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; AND SALAKHUTDINOV, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15 (2014), 1929–1958. doi:10.1214/12-AOS1000. <https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>. (cited on pages 14, 18, 19, and 132)
- SRIVASTAVA, R. K.; GREFF, K.; AND SCHMIDHUBER, J., 2015. Highway Networks. *International Conference on Machine Learning*, (may 2015). <http://arxiv.org/abs/1505.00387>. (cited on page 19)
- STAJNER, S. H. B. AND SAGGION, H., 2015. A Deeper Exploration of the Standard PB-SMT Approach to Text Simplification and its Evaluation. *Annual Meeting of the Association for Computational Linguistics*, (2015), 823–828. doi:10.3115/v1/P15-2135. (cited on pages 125, 126, 137, 138, and 139)
- STAMATATOS, E., 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60, 3 (2009), 538–556. doi:10.1002/asi.21001. (cited on pages 38 and 45)
- SUNDERMEYER, M.; ALKHOULI, T.; WUEBKER, J.; AND NEY, H., 2014. Translation Modeling with Bidirectional Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 14–25. doi:10.3115/v1/D14-1003. <https://www.semanticscholar.org/paper/Translation-Modeling-with-Bidirectional-Recurrent-Sundermeyer-Alkhouli/16fe3d5fc4a671e5b853cb1601700c11a675946ch><http://aclweb.org/anthology/D14-1003>. (cited on page 26)
- SUTSKEVER, I.; VINYALS, O.; AND LE, Q. V., 2014. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, (2014), 3104–3112. (cited on pages 25, 26, 124, 134, 154, and 155)
- SUTTON, R. AND BARTO, A., 1998. Reinforcement Learning: An Introduction. doi:10.1109/TNN.1998.712192. (cited on page 33)
- SWAIN, M. J. AND BALLARD, D. H., 1991. Color indexing. *International Journal of Computer Vision*, 7, 1 (nov 1991), 11–32. doi:10.1007/BF00130487. <http://link.springer.com/10.1007/BF00130487>. (cited on page 13)

-
- SWAYAMDIPTA, S.; THOMSON, S.; DYER, C.; AND SMITH, N. A., 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv Preprint arXiv:1706.09528*, (2017). <https://arxiv.org/abs/1706.09528>. (cited on page 149)
- SZARVAS, G.; BIEMANN, C.; AND GUREVYCH, I., 2013. Supervised All-Words Lexical Substitution using Delexicalized Features. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4, 1 (2013), 1131–1141. <http://www.aclweb.org/anthology/N13-1133>. (cited on page 128)
- SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; AND ALEMI, A., 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *AAAI Conference on Artificial Intelligence*, (2017). <https://arxiv.org/pdf/1602.07261.pdf>. (cited on pages 18 and 20)
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; AND RABINOVICH, A., 2015. Going Deeper With Convolutions. *Computer Vision and Pattern Recognition*, (2015). (cited on pages 5, 11, 18, 20, and 54)
- SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; AND WOJNA, Z., 2016. Rethinking the Inception Architecture for Computer Vision. *Computer Vision and Pattern Recognition*, (2016). doi:10.1109/CVPR.2016.308. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf. (cited on pages 18, 153, 157, 158, and 159)
- TABOADA, M.; BROOKE, J.; TOFILOSKI, M.; VOLL, K.; AND STEDE, M., 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37, 2 (2011), 267–307. doi:10.1162/COLI_a_00049. http://www.mitpressjournals.org/doi/10.1162/COLI_a_00049. (cited on page 99)
- TÄCKSTRÖM, O. AND McDONALD, R., 2011. Discovering Fine-Grained Sentiment with Latent Variable Structured Prediction Models. *Advances in Information Retrieval*, (2011). (cited on page 98)
- TALMY, L., 1975. Figure and Ground in Complex Sentences. In *Annual Meeting of the Berkeley Linguistics Society*, vol. 1, 419–430. (cited on page 60)
- TAUSCZIK, Y. R. AND PENNEBAKER, J. W., 2010. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social*

-
- Psychology*, 29, 1 (2010), 24–54. doi:10.1177/0261927X09351676. <http://journals.sagepub.com/doi/10.1177/0261927X09351676>. (cited on pages 3, 37, 41, 42, and 43)
- TENENBAUM, J. B. AND FREEMAN, W. T., 2000. Separating Style and Content with Bilinear Models. *Neural Computation*, 12, 6 (2000), 1247–1283. doi:10.1162/089976600300015349. <http://www.mitpressjournals.org/doi/10.1162/089976600300015349>. (cited on pages 4, 41, and 42)
- THELWALL, M.; BUCKLEY, K.; PALTOGLOU, G.; CAI, D.; AND KAPPAS, A., 2010. Sentiment Strength Detection in Short Informal Text. *Journal of the Association for Information Science and Technology*, (2010). (cited on pages 99 and 107)
- THOMEE, B.; SHAMMA, D. A.; FRIEDLAND, G.; ELIZALDE, B.; NI, K.; POLAND, D.; BORTH, D.; AND LI, L.-J., 2015. The New Data and New Challenges in Multimedia Research. *arXiv:1503.01817*, (2015). (cited on page 107)
- TIELEMAN, T. AND HINTON, G., 2012. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSE: Neural Networks for Machine Learning. (cited on page 15)
- TITOV, I. AND McDONALD, R., 2008. Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th International Conference on World Wide Web - WWW '08*, 111. ACM Press, New York, New York, USA. doi:10.1145/1367497.1367513. <http://portal.acm.org/citation.cfm?doid=1367497.1367513>. (cited on page 44)
- TOMPSON, J.; GOROSHIN, R.; JAIN, A.; LECUN, Y.; AND BREGLER, C., 2015. Efficient object localization using Convolutional Networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 648–656. doi:10.1109/CVPR.2015.7298664. (cited on page 19)
- TRAN, K.; HE, X.; ZHANG, L.; AND SUN, J., 2016. Rich Image Captioning in the Wild. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference On*, 434–441. IEEE. https://www.cv-foundation.org/openaccess/content_cvpr_2016_workshops/w12/html/Tran_Rich_Image_Captioning_CVPR_2016_paper.html. (cited on pages 84 and 180)
- TU, Z.; LU, Z.; LIU, Y.; LIU, X.; AND LI, H., 2016. Modeling Coverage for Neural Machine Translation. *Annual Meeting of the Association for Computational Linguistics*, (2016), 76–85. doi:10.1145/2856767.2856776. <http://arxiv.org/abs/1601.04811>. (cited on page 143)

-
- TURK, M. M. AND PENTLAND, A. A., 1991. Face Recognition Using Eigenfaces. doi: 10.1109/CVPR.1991.139758. <https://ieeexplore.ieee.org/document/139758/>. (cited on page 13)
- VAN DEN OORD, A.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A.; AND KAVUKCUOGLU, K., 2016. WaveNet: A Generative Model for Raw Audio. *arXiv Preprint arXiv:1609.03499*, (2016). (cited on pages 41 and 42)
- VAN HALTEREN, H.; BAAYEN, H.; TWEEDIE, F.; HAVERKORT, M.; AND NEIJT, A., 2005. New Machine Learning Methods Demonstrate the Existence of a Human Stylome. *Journal of Quantitative Linguistics*, 12, 1 (2005), 65–77. doi:10.1080/09296170500055350. https://www.researchgate.net/profile/Anneke_Neijt/publication/220469157_New_Machine_Learning_Methods_Demonstrate_the_Existence_of_a_Human_Stylome/links/55cb27cd08aea2d9bdcc2465.pdf. (cited on pages 45 and 148)
- VEDANTAM, R.; ZITNICK, C. L.; AND PARIKH, D., 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 4566–4575. doi:10.1109/CVPR.2015.7299087. (cited on pages 34, 36, and 161)
- VENUGOPALAN, S.; ROHRBACH, M.; DONAHUE, J.; MOONEY, R.; DARRELL, T.; AND SAENKO, K., 2015. Sequence to Sequence – Video to Text. In *Proceedings of the IEEE International Conference on Computer Vision*, 4534–4542. (cited on page 25)
- VERDONK, P., 2002. *Stylistics*. Oxford University Press. (cited on page 37)
- VINYALS, O.; FORTUNATO, M.; AND JAITLY, N., 2015a. Pointer Networks. (2015), 1–9. <http://arxiv.org/abs/1506.03134>. (cited on page 27)
- VINYALS, O.; TOSHEV, A.; BENGIO, S.; AND ERHAN, D., 2015b. Show and tell: A neural image caption generator. *Computer Vision and Pattern Recognition*, (2015). (cited on pages 5, 28, 32, 55, 96, 103, 110, 112, 116, 152, 153, 157, and 158)
- VINYALS, O.; TOSHEV, A.; BENGIO, S.; AND ERHAN, D., 2017. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 4 (2017), 652–663. doi:10.1109/TPAMI.2016.2587640. (cited on pages 32, 113, 156, and 180)
- VIOLA, P. AND JONES, M., 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference*

-
- on *Computer Vision and Pattern Recognition*. CVPR 2001, vol. 1, I-511–I-518. doi:10.1109/CVPR.2001.990517. <http://ieeexplore.ieee.org/document/990517/>. (cited on pages 12 and 13)
- VITERBI, A., 1967. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13, 2 (apr 1967), 260–269. doi:10.1109/TIT.1967.1054010. (cited on page 21)
- VORONTSOV, E.; TRABELSI, C.; KADOURY, S.; AND PAL, C., 2017. On orthogonality and learning recurrent networks with long term dependencies. *International Conference on Machine Learning*, (2017). (cited on page 23)
- WAGEMANS, J.; ELDER, J. H.; KUBOVY, M.; PALMER, S. E.; PETERSON, M. A.; SINGH, M.; AND VON DER HEYDT, R., 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological Bulletin*, 138, 6 (2012), 1172. (cited on page 60)
- WAN, L.; ZEILER, M.; ZHANG, S.; LECUN, Y.; AND FERGUS, R., 2013. Regularization of Neural Networks using DropConnect. *International Conference on Machine Learning*, , 1 (2013), 109–111. doi:10.1109/TPAMI.2017.2703082. <http://proceedings.mlr.press/v28/wan13.pdf>. (cited on page 19)
- WANNER, L.; BAEZA-YATES, R.; BRÜGMANN, S.; CODINA, J.; DIALLO, B.; ESCORSA, E.; GIERETH, M.; KOMPATSIARIS, Y.; PAPADOPOULOS, S.; PIANTA, E.; PIELLA, G.; PUHLMANN, I.; RAO, G.; ROTARD, M.; SCHOESTER, P.; SERAFINI, L.; AND ZERVAKI, V., 2008. Towards content-oriented patent document processing. *World Patent Information*, 30, 1 (2008), 21–33. doi:<https://doi.org/10.1016/j.wpi.2007.03.008>. <http://www.sciencedirect.com/science/article/pii/S0172219007000762>. (cited on page 129)
- WEHRLI, E.; SERETAN, V.; AND NERIMA, L., 2010. Sentence Analysis and Collocation Identification. *Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010)*, (2010), 28–36. <https://aclanthology.info/pdf/W/W10/W10-3705.pdf>. (cited on page 148)
- WEST, R. L. AND TURNER, L. H., 2010. *Introducing Communication Theory: Analysis and Application*. ISBN 0767405226. (cited on pages 3 and 40)
- WIDROW, B. AND HOFF, M. E., 1960. Adaptive switching circuits. Technical report, Stanford University CA, Stanford Electronics Labs. (cited on page 15)

-
- WILLIA, R. J., 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8, 3 (1992), 229–256. doi:10.1023/A:1022672621406. (cited on page 33)
- WILLIAMS, R. J. AND PENG, J., 1990. An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories. *Neural Computation*, 2, 4 (1990), 490–501. (cited on page 24)
- WILLIAMS, R. J. AND ZIPSER, D., 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1, 2 (1989), 270–280. doi:10.1162/neco.1989.1.2.270. <http://www.mitpressjournals.org/doi/10.1162/neco.1989.1.2.270>. (cited on page 155)
- WINKE, P.; GASS, S.; AND SYDORENKO, T., 2013. Factors Influencing the Use of Captions by Foreign Language Learners: An Eye-Tracking Study. *The Modern Language Journal*, 97, 1 (2013), 254–275. doi:10.1111/j.1540-4781.2012.01432.x. (cited on page 2)
- WONG, E.; LIU, T.; AND TAN, L., 2015. CloCom: Mining existing source code for automatic comment generation. In *Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference On*, 380–389. IEEE. <https://www.infona.pl/resource/bwmeta1.element.ieee-art-000007081848>. (cited on page 181)
- WONG, K.-F.; WU, M.; AND LI, W., 2008a. Extractive Summarization Using Supervised and Semi-Supervised Learning. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1, COLING '08*, 985–992. Association for Computational Linguistics, Stroudsburg, PA, USA. <http://dl.acm.org/citation.cfm?id=1599081.1599205>. (cited on page 129)
- WONG, M. T.; CHUN, A. H. W.; LI, Q.; CHEN, S. Y.; AND XU, A., 2008b. Automatic Haiku Generation Using VSM. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, 7. World Scientific and Engineering Academy and Society. (cited on page 46)
- WOODSEND, K. AND LAPATA, M., 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 409–420. Association for Computational Linguistics. <https://dl.acm.org/citation.cfm?id=2145480>. (cited on pages 125 and 127)
- WU, D. AND HONG KONG, T., 1995. Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Par-

-
- allel Corpora. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, (1995). http://ccl.pku.edu.cn/doubtfire/nlp/machine{}_translation/template{}_based{}_mt/stochasticinversiontransductiongrammars.pdf. (cited on page 126)
- WU, P. AND DIETTERICH, T. G., 2004. Improving SVM Accuracy by Training on Auxiliary Data Sources. In *Twenty-First International Conference on Machine Learning - ICML '04*, 110. ACM Press, New York, New York, USA. doi:10.1145/1015330.1015436. <http://portal.acm.org/citation.cfm?doid=1015330.1015436>. (cited on page 97)
- WU, Q.; SHEN, C.; LIU, L.; DICK, A.; AND VAN DEN HENGEL, A., 2015. What Value Do Explicit High Level Concepts Have in Vision to Language Problems? (jun 2015). <http://arxiv.org/abs/1506.01144>. (cited on pages 33 and 84)
- WU, S.; WIELAND, J.; FARIVAR, O.; AND SCHILLER, J., 2017. Automatic Alt-text: Computer-generated Image Descriptions for Blind Users on a Social Network Service. *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, (2017), 1180–1192. doi:10.1145/2998181.2998364. (cited on page 2)
- WU, T.-F.; LIN, C.-J.; AND WENG, R. C., 2004. Probability Estimates for Multi-Class Classification by Pairwise Coupling. *Journal of Machine Learning*, 5 (2004), 975–1005. doi:10.1016/j.visres.2004.04.006. <http://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf>. (cited on page 70)
- WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q. V.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K.; KLINGNER, J.; SHAH, A.; JOHNSON, M.; LIU, X.; KAISER, L.; GOUWS, S.; KATO, Y.; KUDO, T.; KAZAWA, H.; STEVENS, K.; KURIAN, G.; PATIL, N.; WANG, W.; YOUNG, C.; SMITH, J.; RIESA, J.; RUDNICK, A.; VINYALS, O.; CORRADO, G.; HUGHES, M.; AND DEAN, J., 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *Arxiv*, (2016). <http://arxiv.org/abs/1609.08144>. (cited on pages 22, 25, 27, 33, and 154)
- WUBBEN, S.; VAN DEN BOSCH, A.; AND KRAHMER, E., 2012. Sentence Simplification by Monolingual Machine Translation. *Annual Meeting of the Association for Computational Linguistics*, , July (2012), 1015–1024. <http://www.aclweb.org/anthology/P12-1107>. (cited on pages 125, 127, 137, 138, and 139)
- XIA, Y.; TIAN, F.; QIN, T.; YU, N.; AND LIU, T.-Y., 2017. Sequence Generation with Target Attention. In *Machine Learning and Knowledge Discovery in Databases*, 816–831. Springer International Publishing, Cham. (cited on page 27)

- XIE, L. AND HE, X., 2013. Picture Tags and World Knowledge: Learning Tag Relations from Visual Semantic Sources. In *ACM Multimedia*. (cited on pages 58, 72, and 152)
- XIE, S.; GIRSHICK, R.; DOLLÁR, P.; TU, Z.; AND HE, K., 2017. Aggregated Residual Transformations for Deep Neural Networks. *Computer Vision and Pattern Recognition*, (2017). <https://arxiv.org/pdf/1611.05431.pdf>. (cited on pages 18 and 20)
- XU, K.; BA, J.; KIROS, R.; COURVILLE, A.; SALAKHUTDINOV, R.; ZEMEL, R.; AND BENGIO, Y., 2015a. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *International Conference on Machine Learning*, (2015). (cited on pages 32, 54, and 96)
- XU, W., 2017. From Shakespeare to Twitter: What are Language Styles all about? *Proceedings of the Workshop on Stylistic Variation*, (2017). (cited on pages 6 and 38)
- XU, W.; CALLISON-BURCH, C.; AND NAPOLES, C., 2015b. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3 (2015), 283–297. (cited on pages 124, 130, 131, 134, and 135)
- XU, W.; NAPOLES, C.; PAVLICK, E.; CHEN, Q.; AND CALLISON-BURCH, C., 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, (2016), 401–415. (cited on pages 125, 127, and 180)
- XU, W.; RITTER, A.; DOLAN, W. B.; GRISHMAN, R.; AND CHERRY, C., 2012. Paraphrasing for Style. *International Conference on Computational Linguistics*, 4, 1234 (2012), 2899–2914. doi:10.1016/j.aqpro.2013.07.003. <http://www.aclweb.org/anthology/C12-1177>. (cited on pages 46 and 161)
- YAGCIOGLU, S.; ERDEM, E.; ERDEM, A.; AND CAKICI, R., 2015. A Distributed Representation Based Query Expansion Approach for Image Captioning. *Annual Meeting of the Association for Computational Linguistics*, (2015), 106–111. <http://www.aclweb.org/anthology/P15-2018>. (cited on pages 28 and 29)
- YAMADA, K. AND KNIGHT, K., 2001. A Syntax-based Statistical Translation Model. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, (2001). <http://www.aclweb.org/anthology/P01-1067>. (cited on page 126)
- YANAI, K. AND BARNARD, K., 2005. Image Region Entropy: A Measure of "Visualness" of Web Images Associated with One Concept. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, 419–422.

-
- ACM, New York, NY, USA. doi:10.1145/1101149.1101241. <http://doi.acm.org/10.1145/1101149.1101241>. (cited on page 152)
- YANG, Y.; TEO, C. L.; DAUMÉ III, H.; AND ALOIMONOS, Y., 2011. Corpus-Guided Sentence Generation of Natural Images. In *Empirical Methods in Natural Language Processing*. (cited on pages 28, 30, 31, and 60)
- YANG, Z.; CHEN, W.; WANG, F.; AND XU, B., 2017a. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. *arXiv*, (2017). (cited on pages 47 and 48)
- YANG, Z.; HU, Z.; SALAKHUTDINOV, R.; AND BERG-KIRKPATRICK, T., 2017b. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. *International Conference on Machine Learning*, (2017). (cited on page 47)
- YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; AND LIPSON, H., 2014. How transferable are features in deep neural networks? <https://dl.acm.org/citation.cfm?id=2969197>. (cited on pages 19, 20, 68, and 98)
- YOU, Q.; JIN, H.; AND LUO, J., 2018. Image Captioning at Will: A Versatile Scheme for Effectively Injecting Sentiments into Image Descriptions. *arXiv Preprint arXiv:1801.10121*, (2018). (cited on pages 53 and 55)
- YOUNG, P.; LAI, A.; HODOSH, M.; AND HOCKENMAIER, J., 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2 (2014), 67–78. (cited on page 32)
- YU, L.; ZHANG, W.; WANG, J.; AND YU, Y., 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *AAAI Conference on Artificial Intelligence*, (2017). doi:10.1001/jamainternmed.2016.8245. <http://www.aaai.org/Conferences/AAAI/2017/PreliminaryPapers/12-Yu-L-14344.pdf>. (cited on pages 47 and 48)
- YUAN, G.-X.; HO, C.-H.; AND LIN, C.-J., 2012. Recent Advances of Large-Scale Linear Classification. *Proceedings of the IEEE*, 100, 9 (sep 2012), 2584–2603. doi:10.1109/JPROC.2012.2188013. (cited on page 72)
- ZAREMBA, W.; SUTSKEVER, I.; AND VINYALS, O., 2014. Recurrent Neural Network Regularization. *arXiv:1409.2329*, (2014). (cited on page 24)
- ZEILER, M. D. AND FERGUS, R., 2014. Visualizing and Understanding Convolutional Networks. *European Conference on Computer Vision*, 8689 (2014), 818–833. doi:10.1007/978-3-319-10590-1_53. https://link.springer.com/chapter/10.1007%2F978-3-319-10590-1_53. (cited on pages 13 and 20)

-
- ZHANG, C.; PLATT, J. C.; AND VIOLA, P. A., 2005. Multiple Instance Boosting for Object Detection. *Neural Information Processing Systems*, 74, 10 (2005), 1769–1775. doi:citeulike-article-id:6144441. <https://papers.nips.cc/paper/2926-multiple-instance-boosting-for-object-detection.pdf>. (cited on page 33)
- ZHANG, L.; SUNG, F.; LIU, F.; XIANG, T.; GONG, S.; YANG, Y.; AND HOSPEDALES, T. M., 2017a. Actor-Critic Sequence Training for Image Captioning. (2017). <https://arxiv.org/pdf/1706.09601.pdf><http://arxiv.org/abs/1706.09601>. (cited on page 33)
- ZHANG, X. AND LAPATA, M., 2014. Chinese Poetry Generation with Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680. (cited on pages 46 and 47)
- ZHANG, X.; ZHAO, J.; AND LECUN, Y., 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, 649–657. <http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica>. (cited on page 16)
- ZHANG, Z.; LIU, S.; LI, M.; ZHOU, M.; AND CHEN, E., 2017b. Stack-based Multi-layer Attention for Transition-based Dependency Parsing. *Empirical Methods in Natural Language Processing*, (2017), 1678–1683. (cited on page 25)
- ZHU, Y.; KIROS, R.; ZEMEL, R.; SALAKHUTDINOV, R.; URTASUN, R.; TORRALBA, A.; AND FIDLER, S., 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. (2015). <http://arxiv.org/abs/1506.06724>. (cited on pages 146, 157, and 159)
- ZHU, Z.; BERNHARD, D.; AND GUREVYCH, I., 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. *International Committee on Computational Linguistics*, (2010), 1353–1361. <http://www.aclweb.org/anthology/C10-1152>. (cited on pages 124, 125, 126, 127, 130, and 138)
- ZILLY, J. G.; SRIVASTAVA, R. K.; KOUTNÍK, J.; AND SCHMIDHUBER, J., 2016. Recurrent Highway Networks. *arXiv Preprint arXiv:1607.03474*, (2016). (cited on page 21)
- ZITNICK, C. L. AND PARIKH, D., 2013. Bringing Semantics Into Focus Using Visual Abstraction. In *Computer Vision and Pattern Recognition*. (cited on page 60)
- ZOPH, B. AND LE, Q. V., 2017. Neural Architecture Search with Reinforcement Learning. *International Conference on Learning Representations*, (2017). <https://research.google.com/pubs/pub45826.html>. (cited on page 24)